TR-EE-65-10
NASAMSFC-HUNTSVILLE, ALABAMA
NASA CONTRACT NUMBER (11415)
PRF 4005

corrected
11416

PURDUE UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING
CONTROL AND INFORMATION SYSTEMS LABORATORY

FINAL REPORT

ADVANCED CONTROL TECHNIQUES APPLIED TO LARGE FLEXIBLE LAUNCH VEHICLES

VOL I

July 1965
LAFAYETTE, INDIANA

J. E. Gibson, Principal Investigator

V. B. Haas
J. C. Hill
L. E. Jones, III
A. S. Morse
S. Murtuza
A. M. Steinberg

N67 19313 (ACCESSION NUMBER)
(PAGES)
(NASA CR OR TMX OR AD NUMBER)
(THRU) None
(CODE) 10
(CATEGORY)
FACILITY FORM 602

Rpt 34307

TR-EE 65-10

FINAL REPORT

Volume I

covering the period
1 July 1964 to 30 July 1965

prepared for

George C. Marshall Space Flight Center, Huntsville, Alabama

National Aeronautics and Space Administration

on

Contract NAS 8-11416

PRF 4005-52-285

titled

ADVANCED CONTROL TECHNIQUES APPLIED TO LARGE FLEXIBLE LAUNCH VEHICLES

with

Control and Information Systems Laboratory

School of Electrical Engineering

Purdue University

Lafayette, Indiana

J. E. Gibson, Principal Investigator

30 July, 1965

Advanced Control Techniques Applied to Large Flexible Launch Vehicles

by

J. E. Gibson

J. C. Hill

V. Haas

A. S. Morse

S. Murtuza

L. E. Jones III

A. Steinberg

July 1965

TABLE OF CONTENTS

VOLUME I

	Page
LIST OF FIGURES	v
ABSTRACT	x
CHAPTER I - INTRODUCTION	1.1
CHAPTER II - THE MINIMAX PROBLEM	2.1
2.1 Introduction	2.1
2.2 The Minimax Problem	2.3
2.3 Min-Max and Max-Min	2.5
2.4 Terminal Conditions	2.8
2.5 Goals for Model Vehicle No. 2	2.9
2.6 The Minimax Principle Formulation	2.10
2.7 An Example	2.13
2.8 Variation of Bounding Conditions	2.17
2.9 Quasilinearization	2.19
2.10 The Ping Pong Algorithm	2.29
2.11 The Gradient Search Formulation	2.32
2.12 Step Size	2.38
2.13 The Gradient Technique for Continuous Systems	2.44
2.14 Problems Investigated	2.49
2.15 Conclusions and Recommendations for Further Study	2.87
References	2.93
CHAPTER III - THE MINIMAX PRINCIPLE	3.1
3.1 Introduction	3.1
3.2 Statement of the Problem	3.1
3.3 The Minimax Principle	3.3
3.4 A Theorem for the Fixed Terminal Time Free Right End Problem	3.9
3.5 Conclusions	3.11
References	3.11
CHAPTER IV - APPLICATION OF WORST CASE ERROR ANALYSIS TO THE DESIGN OF A SPECIFIC OPTIMAL CONTROL FOR MODEL VEHICLE NO. 2	4.1
4.1 Introduction	4.1
4.2 General Problem Statement	4.1
4.3 Worst Case Error Analysis	4.2
4.4 Specific Control for Systems with Disturbances	4.3
4.5 Example	4.3
4.6 Chapter Summary	4.7
References	4.14

Page

CHAPTER V - APPLICATION OF LINEAR PROGRAMMING TO THE OPTIMAL CONTROL OF MODEL VEHICLE NO. 2 . . . . . 5.1

5.1 Introduction 5.1
5.2 Introduction to Linear Programming 5.2
5.3 Solution of Linear Programming Problems 5.5
5.3.1 Variables Not Restricted in Sign 5.6
5.3.2 Inequality Constraints on the Magnitude of a Variable or a Linear Function of the Variables 5.6
5.4 Optimal Control Theory and Linear Programming . . . 5.6
5.4.1 Mathematical Model 5.7
5.4.2 Design Specifications 5.7
5.4.3 Performance Index 5.8
5.5 Modified Optimum Control Policy 5.22
5.6 Treatment of Continuous Systems 5.24
5.7 Computational Results 5.25
5.8 Conclusions and Recommendations 5.64
References 5.70

VOLUME II

CHAPTER VI - ADAPTIVE CONTROL APPLIED TO A FLEXIBLE VEHICLE . 6.1

6.1 Introduction 6.1
6.2 Discussion of the Mathematical Model 6.2
6.3 Conventional Control 6.4
6.3.1 Rigid Body Control 6.4
6.3.2 Flexible Body Control 6.8
6.4 Adaptive Control 6.9
6.4.1 Need and Definition 6.9
6.4.2 Tracking Notch Filter 6.12
6.4.3 Digital Adaptive Filter 6.16
6.4.4 Model Reference Adaptive Control, MRAC . . . 6.20
6.5 Conclusions and Proposed Extensions 6.24
References 6.26

APPENDIX A - ANNOTATED BIBLIOGRAPHY ON ADAPTIVE CONTROL . . . A.1

APPENDIX B - EQUATIONS OF MOTION FOR THE MODEL VEHICLE B.1

APPENDIX C - ALGORITHMS AND SUBROUTINES FOR TWO INTERVAL CONTROL UNDER THE MODIFIED OPTIMAL CONTROL POLICY VIA LINEAR PROGRAMMING C.1

APPENDIX D - A STEEP DESCENT PROCEDURE FOR MINIMIZATION PROBLEMS D.1

APPENDIX E - MINIMUM DRIFT AND MINIMUM LOAD CONTROL E.1

LIST OF FIGURES

Figure		Page
2.1	Variations in Initial and Final Values of $\lambda(t)$ as Functions of Iteration	2.30
2.2	Level Curves for the Performance Index in Equation (2.98) with N=1	2.35
2.3	Solution Curves for Problem 2.1	2.52
2.4	Solution Curves for Problem 2.2	2.53
2.5	Solution Curves for Problem 2.3	2.55
2.6	Solution Curves for Problem 2.4	2.56
2.7a	Solution Curves for Problem 2.5	2.59
2.7b	Solution Curves for Problem 2.5	2.60
2.8a	Solution Curves for Problem 2.6	2.63
2.8b	Solution Curves for Problem 2.6	2.64
2.9a	Solution Curves for Problem 2.7	2.65
2.9b	Solution Curves for Problem 2.7	2.66
2.10a	Solution Curves for Problem 2.8	2.67
2.10b	Solution Curves for Problem 2.8	2.68
2.11	Solution Curves for Problem 2.9	2.72
2.12	Block Diagram Relating β and β_c	2.73
2.13a	Solution Curves for Problem 2.11 Using Worst α_w . .	2.75
2.13b	Solution Curves for Problem 2.11 Using Worst α_w . .	2.76
2.13c	Solution Curves for Problem 2.11 Using M.S.F.C. α_w .	2.77
2.13d	Solution Curves for Problem 2.11 Using M.S.F.C. α_w .	2.78

Figure		Page
2.14a	Solution Curves for Problem 2.12 Using Worst α_w . .	2.80
2.14b	Solution Curves for Problem 2.12 Using Worst α_w . .	2.81
2.14c	Solution Curves for Problem 2.12 Using MSFC α_w . . .	2.82
2.14d	Solution Curves for Problem 2.12 Using MSFC α_w . . .	2.83
2.15	Level Curves Corresponding to a One-Dimensional Minimax Problem with Only a Local Solution	2.84
2.16	Solution Curves for Problem 2.13	2.88
4.1	Bending Moment and Switching Time for Worst Disturbance with a_o as a Parameter, T = 1 sec. . .	4.8
4.2	Bending Moment and Switching Time with a_1 as a Parameter, T = 1 sec.	4.9
4.3	Bending Moment and Switching Time with a_o as a Parameter, T = 5 sec.	4.10
4.4	Worst Disturbance	4.11
4.5	System Response to Worst Disturbance	4.12
4.6	Drift and Bending Moment for Worst Disturbance . . .	4.13
5.1	Response of the System in Example 5.1 with Performance Index: Min $\lvert x_1(K+1) \rvert$ (One-Interval Control)	
	(a) x_1 vs. time, x_2 vs. time.	5.28
	(b) u vs. time.	5.29
5.2	Response of the System in Example 5.1 with P.I.: Min $[\lvert x_1(K+1) \rvert + \lvert x_2(K+1) \rvert]$ (One-interval Control	
	(a) x_1 vs. time, x_2 vs. time	5.30
	(b) u vs. time, P.I. vs. time	5.31
5.3	Response of the System in Example 5.1 with P.I.: Min $\lvert x_1(K+2) \rvert$ (Two-interval Control)	
	(a) x_1 vs. time, x_2 vs. time.	5.32
	(b) u vs. time.	5.33

Figure		Page
5.4	Response of the System in Example 5.1 with P.I.: Min $\lvert x_1(K+2) \rvert$ under Modified Optimal Control Policy (MOCP)[1]	
	(a) x_1 vs. time, x_2 vs. time.	5.34
	(b) u vs. time.	5.35
5.5	Response of the System in Example 5.1 with P.I.: Min $[\lvert x_1(K+2) \rvert + \lvert x_2(K+2) \rvert]$ under MOCP	
	(a) x_1 vs. time, x_2 vs. time.	5.37
	(b) u vs. time, P.I. vs. time.	5.38
5.6	Ho and Brentani Solution to Example 5.2 with the Constraint $\lvert u \rvert \leq 1$.	5.41
5.7	Solution to Example 5.2 under MOCP with the Constraint $\lvert u \rvert \leq 1$.	
	(a) x_1 vs. time, x_2 vs. time, x_3 vs. time. . . .	5.42
	(b) x_4 vs. time, u vs. time.	5.43
	(c) P.I. vs. time.	5.44
5.8	Ho and Brentani Solution to Example 5.2 with the Constraint $\lvert x_4 \rvert \leq 1$.	5.45
5.9	Solution to Example 5.2 under MOCP with the Constraint $\lvert x_4 \rvert \leq 1$.	
	(a) x_1 vs. time, x_2 vs. time, x_3 vs. time. . . .	5.46
	(b) x_4 vs. time, u vs. time.	5.47
	(c) P.I. vs. time.	5.48
5.10	Response of MV2 (rigid, time-invariant) under MOCP with Performance Index (5.68) and State and Control Variables Constraint (5.69).	
	(a) ϕ vs. time, $\dot{\phi}$ vs. time, α vs. time.	5.51
	(b) β vs. time.	5.52
	(c) $\dot{Z}$ vs. time, Z vs. time.	5.53
	(d) B.M. vs. time.	5.54
5.11	Response of MV2 (rigid, time-invariant) under MOCP with Performance Index (5.71) and Constraint (5.70)	
	(a) ϕ vs. time, $\dot{\phi}$ vs. time, α vs. time.	5.55
	(b) β vs. time.	5.56
	(c) $\dot{Z}$ vs. time, Z vs. time.	5.57
	(d) B.M. vs. time.	5.58

Figure		Page
5.12	Response of MV2 (rigid, time-invariant) under MOCP with Performance Index (5.76) and Constraint (5.70).	
	(a) ϕ vs. time, $\dot{\phi}$ vs. time, α vs. time.	5.60
	(b) β vs. time.	5.61
	(c) $\dot{Z}$ vs. time, Z vs. time.	5.62
	(d) B.M. vs. time.	5.63
5.13	Response of MV2 (rigid, time-varying) under MOCP with Performance Index (5.68) and Constraints (5.69)	
	(a) α_w vs. time, ϕ vs. time, $\dot{\phi}$ vs. time, α vs. time, β vs. time.	5.65
	(b) $\dot{Z}$ vs. time, Z vs. time, B.M. vs. time. . . .	5.66
5.14	Computer-Controlled Closed-Loop System	5.68
6.1	A Tracking Notch Filter	6.15
6.2	A Frequency Tracking Unit - A Self-Adjusting System	6.15
6.3	A Model Reference Adaptive Control System	6.21
B.1	Definition of Symbols	B.2,3
B.2	Rigid Body Coordinate System - Pitch Plane	B.4
B.3	Variation of C_1 With Time	B.7
B.4	Variation of C_2 With Time	B.8
B.5	Variation of C_3 With Time	B.9
B.6	Variation of C_4 With Time	B.10
B.7	Variation of C_5 With Time	B.11
B.8	Variation of C_7 With Time	B.12
B.9	Wind Inputs for Pitch and Yaw Planes	B.14
B.10	Flexible Body Coordinate System	B.15

Figure		Page
E.1	Root Locus for $a_o = 0$, $b_o = 0$, a_1 Increasing	E.6
E.2	Root Locus for $a_o = 0$, $a_1 = 2.5$, b_o Increasing . . .	E.7
E.3	Root Locus for $a_o = 0$, $a_1 = 0$, b_o Increasing	E.8
E.4	Response of MV2 under Minimum Load Control	E.11
E.5	Response of MV2 under Minimum Drift Control	E.12

ABSTRACT

This is the final report of a one year study by the Control and Information Systems Laboratory (CISL), Purdue University for Marshall Space Flight Center (MSFC), NASA on the application of advanced control techniques to large flexible launch vehicles. The report consists of six chapters and five appendices, titled: I, Introduction; II, The Minimax Problem; III, The Minimax Principle; IV, Application of Worst Case Error Analysis to the Design of a Specific Optimal Control for Model Vehicle No. 2; V, The Application of Linear Programming to the Optimal Control of Model Vehicle No. 2; VI. Adaptive Control Applied to a Flexible Vehicle; Appendix A, An Annotated Bibliography on Adaptive Control; Appendix B, Equations of Motion for the Model Vehicle; Appendix C, Algorithms and Subroutines for the Solution of Linear Programming Problems; Appendix D, A Steep Descent Procedure for Minimization Problems; Appendix E, Minimum Drift and Minimum Load Control.

Attempts are made to bring a wide variety of modern optimal control techniques to bear on the problem of controlling a large flexible launch vehicle. Some solutions are found for simplified models and a number of difficulties are pointed out. Recommendations for further study are given.

CHAPTER I

INTRODUCTION

This report discusses the results of a one-year investigation of the applicability of modern theoretical techniques to the problem of controlling a large flexible launch vehicle. Older vehicles and even present-day boosters are sufficiently stiff that the natural modes of vibration are at frequencies high with respect to the band pass of the control and guidance systems. Thus it is possible to suppress transients induced by these modes by simple techniques such as low-pass or band-elimination filtering. As future vehicles are increased in size such techniques will fail. It will be necessary to employ advanced techniques which do not combat by brute force the tendency of a large vehicle to bend. A portion of this contract was devoted to the application of modern control techniques to the solution of this problem.

Even before the problem of controlling a realistic, flexible vehicle is encountered, a solution to a rather complex control problem for the simpler rigid vehicle must be found. This is the problem of finding an optimum control technique for a rigid vehicle which will be flown through regions of wind perturbations which may act to drive the vehicle off course. Since it is desired to <u>minimize</u> some suitably defined error function for a worst wind, or <u>maximum</u> perturbation condition, this is called the <u>Minimax problem</u>. The particular wind which causes this worst case should also be determined by the solution.

It was impressed upon CISL by MSFC that primary attention should be addressed to this problem before going on to the apparently even more difficult case of a flexible vehicle.

In response to the order of priorities established by MSFC the greater attention was given to the Minimax problem. Chapters II, III, IV and V are devoted to this problem while Chapter VI is designed to lead into the problem of controlling a flexible vehicle.

Chapter II gives a technical introduction to the Minimax Problem including a discussion of the related min-max and max-min problems. A Minimax principle which is derived in Chapter III is applied to the problem. Various computational schemes used in conjunction with this principle are then treated. Methods considered include a form of approximation in policy space (Ping Pong) and Quasilinearization. Solution of the Minimax Problem via gradient methods is also discussed. Thirteen example problems are solved and the efficiency of the various computational schemes on these problems is indicated. An attempt is made to assess the practical importance of the class of solutions presently available for the Minimax Problem.

Chapter III consists of the proofs of several theorems on necessary conditions which a solution to a minimax problem must meet. These proofs were offered in abbreviated form in the eleventh monthly progress report, however, it was subsequently discovered that this report contained several typographical errors. Thus the present treatment should be followed where the two do not agree.

In Chapter IV a design technique is presented in which a certain generality is traded for improved ease in solution. The following point of view is taken. Since even if a Minimax problem could be solved in

general, the probability is that the form of the required controller would be excessively complex. Why not specify a priori the form of a practical controller and seek its optimum parameter values? It is illustrated that this restriction in form to a Specific Optimal Controller materially aids in obtaining solutions.

Chapter V discusses the formulation of the optimization problem with state- and control-variable constraints in terms of Linear Programming (LP) and presents solutions. It is found here that a conventional LP formulation for a rigid body approximation to MV2 would exceed the capacity of present-day digital computers (IBM 7094). Thus a novel per-interval type of control policy is proposed and shown to yield computational results which compare quite favorably with results obtained by conventional optimization schemes. The new approach with LP requires no initial guess for the solution and thus often demands less computer time than the conventional methods; moreover it appears that the new method provides solutions for problems which do not yield to the conventional approach.

Chapter VI consists of a general discussion of various adaptive control schemes as they might apply to MV2 and suggestions for future work. Five appendices complete the report.

By far the great majority of work in modern optimal control fails to grapple with the multitude of very real problems which lie beyond mere formulation and formal, symbolic manipulation. Thus it is not surprising that the Minimax Problem, which is at least an order of magnitude more difficult than ordinary minimization problems, has yielded only scattered results in a single year's work. A continuing effort is essential if major gains are to be realized. Thus this report attempts to outline in

some detail areas for future work. Since this project team is being disbanded no renewal will be sought at this time by CISL. However, individual workers retain a keen interest in making further efforts. Doubtless, the impetus given by MSFC to the profession to seek solutions of the minimax problem, by this and other contracts, will eventually yield direct and lasting benefits to the National space program.

CHAPTER II

THE MINIMAX PROBLEM

2.1 Introduction

In the design of control systems, one is often faced with the problem of building a controller which will satisfactorily regulate a plant in the presence of disturbance inputs. Often the designer knows little about the nature of the disturbances which will be encountered.

One approach to this problem is to design a controller which will adapt itself to the actual disturbance as it is encountered during operation. This approach, which is called "input adaptive control" [1], requires no a priori knowledge about the disturbance input.

Sometimes however, the designer does know something about the type of disturbance which will be encountered. If the statistical nature of the disturbance is known, the designer might use the method developed by Newton, et.al.[2], which produces an optimal controller with respect to a specified preformance index.

Unfortunately, in many instances the statistical properties of a disturbance are not available. However, one usually does have a feel for the range of possible values of the disturbance (i.e., the bounds on the disturbance). For example, in the attitude control problem for a space booster, information about reasonable values of the disturbance α_w (wind angle) is available. Thus a design scheme based upon bounds on disturbance inputs appears to be of value.

Such an approach has been taken by Howard [3], and is called "Worst Case Error Analysis."

Using this approach, the designer can find the worst possible disturbance (in terms of a specified performance index) for a given plant and controller. Using this information, a better controller can then be designed. Having found the new controller, Worst Case Error Analysis can again be applied to find a new worst disturbance. Continuing in this way, hopefully one will end up with a controller which is the best possible for the worst distrubance, the worst disturbance having been calculated for the system using this best controller. In essence, this is the Minimax Problem, the subject of this chapter.

In section 2.2 the Minimax Problem is formulated. Following this are discussions of various aspects of the problem including the choice of performance index and terminal conditions.

The main purpose of the research which has led to the results appearing in this chapter has been to find ways of solving the Minimax Problem. Two methods are proposed. The first involves the direct application of the Minimax Principle which is put on firm mathematical grounds in Chapter III. In section 2.7 the principle is applied to a simple example and an analytic solution is obtained. The example brings to light several interesting properties of the Minimax Problem.

In general, application of the Minimax Principle leads to complicated two-point boundary value problems which cannot easily be solved analytically. Thus, several computational schemes for solving boundary value problems are discussed including Quasilinearization [4] and the Ping Pong algorithm. The investigation of Quasilinearization in section 2.9 leads to several modifications of the scheme which make it computationally more efficient in terms of computer time and memory.

The second method investigated for solving the Minimax Problem is a Gradient Technique [5], [6], [7], [8], [9], which is discussed in section 2.11. The main difficulty encountered in applying the Gradient Technique is the determination of suitable step sizes. Two methods for generating step sizes are discussed in section 2.12.

The Gradient Technique is compared with the Ping Pong algorithm when the latter is used in conjunction with the Minimax Principle. The results of this comparison, indicated in section 2.13, show that the two schemes are quite similar.

The various computational methods discussed in this chapter have been applied to a number of Minimax Problems. These problems and their solutions are presented in section 2.14.

Finally, in section 2.15, important results of the chapter are summarized and recommendations for further study indicated.

2.2 The Minimax Problem

The discussion starts with a formal statement of the Minimax Problem. Consider the vector differential equation

$$\dot{x} = f(x, t, u) + g(x, t, v) \tag{2.1}$$

which describes a dynamical system. x is an n-vector describing the state of the system. f and g are n-vectors and t, u and v are scalar time functions. u is considered to be the control signal while v is the disturbance. It is assumed that the initial conditions

$$x(0) = x_0 \tag{2.2}$$

are given. The performance index

$$J(u, v) = \int_0^T [F(x, t, u) + G(x, t, v)]\, dt + h(x(T), T) \tag{2.3}$$

is also specified, where F, G and h are scalar functions. The boundary conditions

$$\tau(x(T), T) = 0 \tag{2.4}$$

may be given in some cases. τ is an m-vector where $m \leq n$. The additional "in flight" constraints

$$\begin{aligned} \psi^u(u(t)) &\leq 0 \\ \psi^v(v(t)) &\leq 0 \end{aligned} \tag{2.5}$$

may also be specified.

With reference to the above conditions, the Minimax Problem is stated as follows: "Determine the control $u^*(t)$ and the disturbance $v^*(t)$ such that J(u, v) defined in equation (2.3) satisfies

$$J(u^*, v) \leq J(u^*, v^*) \leq J(u, v^*) \tag{2.6}$$

for all admissible u(t) and v(t)."

It is worth emphasizing the meaning of equation (2.6). This equation says that if v^* is given, u^* will provide best performance (smallest value of J) and if u^* is given, v^* will provide worst performance (largest value of J). Equation (2.6) does <u>not</u> say that u^* will provide best performance for any v or that v^* will provide worst performance for any u. For example, if v is not v^*, it may well be true that a smaller J can be obtained with some u other than u^*. This means that it is possible that

$$J(u, v) \leq J(u^*, v) \tag{2.7}$$

This point throws a shadow on the meaning of the Minimax Problem. Of what value is it to operate a practical system with the best control for the worst distrubance if the worst disturbance is not the one actually encountered during operation? A possible answer to this question is as follows. Even if the worst disturbance is not encountered,

operation with the best control will produce results no worse (in terms of performance index) than those which would occur if the disturbance was the worst possible. Thus it is possible to put an upper bound on the value of the performance index when the best control is used. Such information could serve as a guide in the design of an actual control system.

There is another difficulty associated with the Minimax Problem. Suppose that there are two different (u^*, v^*) pairs satisfying equation (2.6). A question arises as to which pair is the desired one. A similar problem arises in a minimization problem when there are two local minima. Though in a minimization problem the question can be resolved by picking the lowest minimum, no simple answer exists for the Minimax Problem. Since in the Minimax Problem, minimization and maximization are performed simultaneously,[1] there is no criterion to distinguish between two pairs of solutions. However, if one is interested in obtaining a least upper bound on the performance index (as suggested above) then the (u^*, v^*) pair giving the lowest valued performance index should be chosen.

2.3 Min-Max and Max-Min

The form of equations (2.1) and (2.3) merit some further discussion. Note that both expressions are written as separable functions of u and v(i.e., no cross-product terms). To understand why they are written this way, it will be helpful to discuss briefly the min-max and max-min problems.

In a min-max problem, one tries to find a u^* and a v^* such that

$$J(u^*, v^*) = \min_u \max_v J(u, v) \tag{2.8}$$

---

1) See the last paragraph in section 2.3.

This "minmax" notation has the following meaning. First $J(u, v)$ is maximized with respect to v. This leads to

$$v^* = v^*(x(t), t, u) \tag{2.9}$$

and from equation (2.8)

$$J(u^*, v^*) = \min_u J(u, v^*(x(t), t, u)) \tag{2.10}$$

The performance index in equation (2.10) is then minimized with respect to u. Thus

$$u^* = u^*(x(t), t) \tag{2.11}$$

It is clear that in a min-max problem, the performance index is first maximized with respect to v and then minimized with respect to u.

In a max-min problem, these operations are reversed. For the max-min problem it follows that

$$J(u^*, v^*) = \max_v \min_u J(u, v) \tag{2.12}$$

$$u^* = u^*(x(t), t, v) \tag{2.13}$$

$$J(u^*, v^*) = \max_v J(u^*(x(t), t, v), v) \tag{2.14}$$

$$v^* = v^*(x(t), t) \tag{2.15}$$

When the u^*, v^* and $J(u^*, v^*)$ in equations (2.11), (2.9) and (2.8) are the same as those in equations (2.13), (2.15) and (2.12), the min-max and max-min solutions are the same. It is clear from equation (2.6) that for a problem to have a Minimax solution, its min-max and max-min solutions must agree. Isaacs[1)] has referred to this condition as the Minimax Assumption. It will be shown in Chapter III that the Minimax Assumption is valid for problems with dynamic equations and performance indices which are separable in control and disturbance. Of course, separability is only a sufficient condition and therefore may be overly

---

1) See [10], p. 35.

restrictive. For example, Berkovitz [11], [12] has investigated problems which are not restricted in this way.

In this investigation of the Minimax Problem, only problems with separable functions (i.e., the problem formulated in section 2.2) are considered. There is physical justification for this restriction. First, the dynamic equations for Model Vehicle No. 2 are indeed separable. Second, there seems to be no physical reason for explicitly placing v (α_w) in the performance index, much less a cross-product of v and u (β).

It is true however, that cross-product terms could enter the performance index indirectly as evidenced by problem 2.10 in section 2.14. If the integral-squared value of bending moment is used as a performance index, the product of α_w and β appears. Thus further investigation of the Minimax Problem with nonseparable functions has some justification on physical grounds.

Before leaving this section, it is worth while to clarify a point of possible confusion. In the preceding discussion it has been indicated that the order in which the min and max operations are performed can affect the solution to the problem. In a Minimax Problem these operations must be interchangeable. This implies that the performance index is to be simultaneously minimized with respect to u and maximized with respect to v. Yet minimization and maximization need not be performed simultaneously when computing solutions. For example, one might attempt to solve a Minimax Problem as follows: Choose a v(t) and minimize the performance index with respect to u(t). Using the resulting u(t), then maximize the performance index with respect to v(t). Using this v(t), compute a new u(t) which minimizes the performance index. Continue the

process until u(t) and v(t) no longer change. If this condition is attained, it is clear from equation (2.6) that the Minimax Problem has been solved even though u(t) and v(t) have not been computed simultaneously.

2.4 Terminal Conditions

In section 2.2, the Minimax Problem has been formulated with fixed terminal time. Though a free terminal-time problem could also be formulated, the fixed terminal-time problem seems more meaningful for Model Vehicle No. 2, because the trajectory flight time is fixed at 157 seconds. Thus, only the fixed terminal time problem is considered.

At the terminal time T, one might require the state of the system to satisfy some condition as indicated in equation (2.4). Usually this condition is that some or all of the state variables take on fixed values at the final time. With the exception of problem 12 (see section 2.14) problems of this type have not been considered. There are several reasons for this. The first is because such problems are quite difficult to solve. It is difficult enough to obtain solutions to the Minimax Problem with free final states. Fixed final states add another degree of complexity to the problem.

Another reason for not using fixed final states is that such a formulation has little meaning in a Minimax Problem. For example,[1] suppose that it is desired to have $\dot{\phi}(T) = 0$. Assume that a β^* and a α_w^* have been found (subject to $\dot{\phi}(T) = 0$) which minimize and maximize respectively, some specified performance index. If the vehicle (using β^* as the control) actually encounters some α_w other than α_w^*, $\dot{\phi}(T)$ will not in general be zero. Thus the worst α_w produces the most desirable value of $\dot{\phi}(T)$ (zero), while α_w's other than α_w^* produces less desirable values

---

1) In this example, the plant is Model Vehicle No. 2.

($|\dot{\phi}(T)| > 0$). Because of this, little physical significance can be attached to α_w^* and thus to the fixed final-state Minimax Problem.

A more reasonable formulation of the problem is as follows. Let $\dot{\phi}(T)$ be free and add $\dot{\phi}^2(T)$ to the performance index. The weighting of this term in the performance index should depend on the importance of having $\dot{\phi}(T) = 0$ relative to other desired goals (see section 2.5). If it is not too important to have $\dot{\phi}(T) = 0$, then $\dot{\phi}^2(T)$ should be weighted lightly. On the other hand, if this condition is very important $\dot{\phi}^2(T)$ should be weighted heavily. Assume that $\dot{\phi}(T) = 0$ is so important that the performance index is just $\dot{\phi}^2(T)$. Let (β^*, α_w^*) be a solution to the associated Minimax Problem. If the vehicle (using β^* as the control) encounters some α_w other than α_w^*, the value of $\dot{\phi}^2(T)$ will be smaller than the value produced with the worst α_w. Thus the worst α_w now produces the most undesirable value of $\dot{\phi}(T)$. Using this problem formulation, the worst α_w has a physical interpretation.

2.5 Goals for Model Vehicle No. 2

At this point it is worthwhile to introduce the idea of a "goal" and to distinguish between it and a performance index. By a goal is meant a word statement of a condition which we would like to see exist in a particular problem. For example, a goal for Model Vehicle No. 2 might be to keep the attitude rate as small as possible throughout flight. It is clear (1) that this is a desirable condition and (2) that it is not a precise statement. For example, by making attitude rate as small as possible at one instant of time may cause it to be large at another. Clearly a more precise statement is needed. The more precise statement is called a performance index. Specifically, a performance index is a suitable mathematical expression which if extremized will in some sense

cause the desired goal to be attained. By a suitable performance index is meant one which can be extremized by existing mathematics. Considering the aforementioned goal for Model Vehicle No. 2, a suitable performance index might be the integral-squared value of attitude rate.

For Model Vehicle No. 2 there seem to be three goals:

1. Keep the vehicle reasonably close to the nominal trajectory all the time (i.e., keep attitude, attitude rate, and drift small).
2. Make attitude, attitude rate and drift as small as possible at the end of the first stage boost (157 seconds).
3. Keep the bending moment well within the structural limits of the vehicle.

It should be recognized that one of the most difficult problems associated with optimization theory is the choosing of a suitable performance index which is representative of the desired goals. There is some discussion of this problem in section 2.14.

Having discussed the Minimax Problem, we now seek to solve it. There are at least three different paths one might take: Differential Game Theory, the Minimax Principle and Gradient Techniques. The first of these approaches is discussed in [10] to [12] and will not receive further consideration in this report. In the following sections both the Minimax Principle and a Gradient Technique will be discussed and their relation indicated.

2.6 The Minimax Principle Formulation

In the work that follows, linear, time-invariant plants are investigated. The study of linear plants is justified since the attitude equations for Model Vehicle No. 2 are linear (to a first approximation). It should also be recognized that all of the following

equations would remain the same for linear, time-varying plants.[1] Under these assumptions the Minimax Problem is formulated as follows:

1. The plant to be controlled is described by a set of n first-order differential equations of the form

$$\dot{x} = Ax + Bu + Cv \tag{2.16}$$

where x is an n-vector describing the state of the plant

A is a constant n x n matrix

B and C are constant n-vectors

u and v are scalar functions of time, u being the control input and v being the disturbance input.

2. The initial conditions

$$x(0) = x_o \tag{2.17}$$

are assumed to be known.

3. The disturbance v is restricted to lie between fixed bounds. A similar restriction is placed on the control u in some problems.

$$V_1 \leq v \leq V_2 \tag{2.18a}$$

$$U_1 \leq u \leq U_2 \tag{2.18b}$$

4. The terminal time T is fixed at some given value.

5. The values of the state variables at the terminal time are free.

6. The performance index[2] is of the form

$$J(u,v) = \frac{1}{2}\int_0^T [x'Qx + au^2 + 2b'x]dt + \frac{1}{2}x'(T)Px(T) \tag{2.19}$$

where Q and P are constant, n x n positive-semidefinite matrices

b is a constant n-vector

a is a non-negative scalar

---

1) For the time varying case A,B,C, V_1,V_2, U_1 and U_2 in equations (2.16) to (2.31) would be function of time rather than constants.

2) For an interpretation of this performance index in terms of physical goals, see section 2.14.

7. It is desired to find u^* and v^* such that

$$J(u^*, v) \leq J(u^*, v^*) \leq J(u, v^*) \tag{2.20}$$

for all admissible u and v.

The above seven statements specify the Minimax Problem. In applying the Minimax Principle (see Chapter III) to this problem one first forms the Hamiltonian function

$$H(x, \lambda, u, v) = \frac{\lambda_o}{2} [x'Qx + au^2 + 2b'x] + <\lambda, (Ax + Bu + Cv)> \tag{2.21}$$

The brackets < > denote inner product and the prime denotes transpose. It is assumed that $\lambda_o \neq 0$ and it is taken as

$$\lambda_o = 1 \tag{2.22}$$

By taking

$$\dot{x} = \nabla_\lambda H \tag{2.23}$$

equation (2.16) is obtained. $\nabla_\lambda H$ is the gradient of H with respect to λ. Taking

$$\dot{\lambda} = - \nabla_x H \tag{2.24}$$

gives

$$\dot{\lambda} = - Q'x - A'\lambda - b \tag{2.25}$$

From the transversality condition it follows that

$$\lambda(T) = \lambda_o P'x(T) \tag{2.26}$$

since T is fixed and the final values of the state variables are free. The Minimax Principle also requires that

$$H(x^*, \lambda^*, u^*, v) \leq H(x^*, \lambda^*, u^*, v^*) \leq H(x^*, \lambda^*, u, v^*) \tag{2.27}$$

where x^* and λ^* describe the optimal trajectory. Note that equation (2.27) must hold for all admissible u and v and for each instant of time between t = 0 and t = T. From equations (2.18a), (2.21) and (2.27) it is clear that

$$v^* = \{ \begin{matrix} V_2 & \text{if} & < \lambda^*, C > > 0 \\ V_1 & \text{if} & < \lambda^*, C > < 0 \end{matrix} \} \tag{2.28}$$

Thus v^* is bang-bang. Since

$$\frac{\partial H}{\partial u} = au + < \lambda, B > \tag{2.29}$$

the optimal control is

$$u^* = \left\{ \begin{matrix} U_2 & \text{if} & U_2 < - \dfrac{< \lambda^*, B >}{a} \\ - \dfrac{< \lambda^*, B >}{a} & \text{if} & U_1 \leq - \dfrac{< \lambda^*, B >}{a} \leq U_2 \\ U_1 & \text{if} & U_1 > - \dfrac{< \lambda^*, B >}{a} \end{matrix} \right\} \tag{2.30}$$

if $a \neq 0$ or

$$u^* = \left\{ \begin{matrix} U_2 & \text{if} & < \lambda^*, B > < 0 \\ U_1 & \text{if} & < \lambda^*, B > > 0 \end{matrix} \right\} \tag{2.31}$$

if $a = 0$. Thus if $a = 0$, the optimal control, u^*, is also bang-bang. The above equations are necessary conditions for the solution of the Minimax Problem.

2.7 An Example

The following example is intended to indicate how the Minimax Principle is used. Consider the first-order plant described by

$$\dot{x} = u + v \tag{2.32}$$

with initial condition

$$x(0) = x_o \tag{2.33}$$

It is required that

$$\begin{matrix} |u| \leq 2 \\ |v| \leq 1 \end{matrix} \tag{2.34}$$

The final time T is fixed and the performance index is

$$J(u, v) = \frac{1}{2} x^2(T) \tag{2.35}$$

Using the Minimax Principle, the Hamiltonian is

$$H(x, \lambda, u, v) = \lambda(u + v) \tag{2.36}$$

The adjoint equation is

$$\dot{\lambda} = 0 \tag{2.37}$$

From the transversality condition

$$\lambda(T) = \lambda_o \, x(T) \tag{2.38}$$

Assume first that a solution exists for $x^*(T) = 0$. Then from (2.35), the performance index $J(u^*, v^*) = 0$. Now choose some $v \neq v^*$ and $u = u^*$ such that the corresponding $x(T) \neq 0$. Inspection of (2.32) indicates that this can always be done. Then $J(u^*, v) > 0$ and what's more important, $J(u^*, v) > J(u^*, v^*)$. But this contradicts equation (2.20). Therefore the hypothesis that a solution exists for $x^*(T) = 0$ is false.

Now assume that $x^*(T) \neq 0$. Then from equation (2.38), for a solution to exist $\lambda_o \neq 0$. Choose

$$\lambda_o = 1 \tag{2.39}$$

Inspection of equation (2.36) indicates that H will satisfy (2.27) if

$$u^*(t) = - 2 \operatorname{sgn} \lambda^*(t) = - 2 \operatorname{sgn} x^*(T) \tag{2.40}$$

and

$$v^*(t) = \operatorname{sgn} \lambda^*(t) = \operatorname{sgn} x^*(T) \tag{2.41}$$

Suppose $x^*(T) > 0$. Then from equations (2.32), (2.40) and (2.41)

$$\dot{x} = -1 \tag{2.42}$$

Using equation (2.42) it follows that

$$x^*(T) = x_o - T \tag{2.43}$$

But it has been assumed that $x^*(T) > 0$. Therefore, in order to satisfy the Minimax Principle

$$x_o - T > 0 \quad \text{if} \quad x^*(T) > 0 \tag{2.44}$$

By similar reasoning

$$x_o + T < 0 \quad \text{if} \quad x^*(T) < 0 \tag{2.45}$$

From (2.44) and (2.45) it can be seen that

$$|x_o| > T \tag{2.46}$$

must hold if the Minimax Principle is to be satisfied. Since the Minimax Principle gives necessary conditions, if (2.46) does not hold no solution exists.

Let us go further. Assuming equation (2.44) holds, from (2.40) and (2.41)

$$\left.\begin{aligned} u^* &= -2 \\ v^* &= 1 \end{aligned}\right\} \tag{2.47}$$

and

$$J(u^*, v^*) = \frac{(x_o - T)^2}{2} \tag{2.48}$$

Now consider the performance index evaluated with the best control u^* and with some admissible v other than v^*. For example

$$J(u^*, v)\Big|_{v = -1} = \frac{(x_o - 3T)^2}{2} \tag{2.49}$$

From equation (2.20)

$$J(u^*, v) \leq J(u^*, v^*) \tag{2.50}$$

for all admissible v. Using (2.48) to (2.50)

$$\frac{(x_o - 3T)^2}{2} \leq \frac{(x_o - T)^2}{2} \tag{2.51}$$

or

$$x_o^2 - 6x_oT + 9T^2 \leq x_o^2 - 2x_oT + T^2 \tag{2.52}$$

and finally

$$x_o \geq 2T \tag{2.53}$$

If (2.45) holds instead of (2.44), similar reasoning leads to

$$-x_o \geq 2T \tag{2.54}$$

Combining (2.53) and (2.54) it can be concluded that no Minimax solution exists for this problem unless

$$|x_o| \geq 2T \tag{2.55}$$

The above example is quite simple but it is certainly not a "gimmick" problem. In fact, it is similar in form to the type of problems encountered with Model Vehicle No. 2. Yet even this simple example has exhibited some of the difficult problems associated with obtaining Minimax solutions. The following observations are made:

1. If equation (2.46) is not satisfied, there is no solution to this problem. It is reasonable to expect that this situation could occur in more complicated problems.
2. If equation (2.46) is satisfied, the conditions of the Minimax Principle are satisfied. However, if equation (2.55) is not satisfied, there is no solution to the problem. Therefore solutions satisfying the Minimax Principle are not necessarily solutions to the problem. This is because the Minimax Principle provides only necessary conditions.

In solving more complicated problems, one must rely on the output of a computer for answers. If the algorithm used to solve a particular problem does not provide a convergent solution, one must decide if it is the fault of the algorithm or if no solution to the problem exists. This is usually a difficult task.

Besides the difficulties indicated thus far, when the Minimax Principle is used, one is faced with the task of solving a nonlinear two-point boundary

value problem. Equations (2.16) and (2.25) with boundary conditions (2.17) and (2.26) describe the problem. The nonlinearity (even for a linear plant with a quadratic performance index) comes from the nature of u^* and v^* described in (2.28), (2.30) and (2.31). It is true that these three equations come from the "hard" constraints in equation (2.18). However, without such constraints (at least (2.18a)) v^* would be unbounded making the problem meaningless. In the following sections, several computational algorithms used to solve such problems will be discussed.

2.8 Variation of Boundary Conditions

For the sake of clarity, the specific boundary value problem which is to be solved is repeated here. Since u^* and v^* are functions of λ, it is possible to write the plant equation as

$$\dot{x} = M(x, \lambda) \tag{2.56}$$

where M is a linear function of x and a nonlinear function of λ. The adjoint equation is

$$\dot{\lambda} = N(x, \lambda) \tag{2.57}$$

N is a linear function of both x and λ. From (2.17) and (2.26) the boundary conditions are

$$x(0) = x_o \tag{2.58}$$

and

$$\lambda(T) = P'x(T) \tag{2.59}$$

These four equations completely specify the boundary value problem to be solved.

One of the most obvious ways to try to solve the boundary value problem is described in the following algorithm:

1. Guess an initial condition vector for λ and call it $\lambda^{o}(0)$.

2. Set $m = 0$.

3. Set $x^{m+1}(0) = x_{o}$

   $\lambda^{m+1}(0) = \lambda^{m}(0)$

   $m = m+1$

   and solve (2.56) and (2.57) for $x^{m}(T)$ and $\lambda^{m}(T)$ by integrating.

4. Define the error

   $$\epsilon^{m} = || \lambda^{m}(T) - P'x^{m}(T) ||$$

5. If $\epsilon^{m} \leq$ TEST go to 8. (TEST is a small preassigned number)

   If $\epsilon^{m} >$ TEST go to 6.

6. Set

   $$\lambda^{m}(T) = P'x^{m}(T)$$

   and solve for $\lambda^{m}(0)$ by integrating (2.56) and (2.57) backwards.

7. Go to 3.

8. Stop, problem completed.

In the above algorithm, m denotes the number of iterations. At the end of each iteration, new initial conditions for λ are generated as indicated in 3. If the error ϵ^{m} becomes sufficiently small, it is clear that the desired boundary conditions will be closely satisfied. A solution to problem 2.13 (see section 2.14) was obtained using this algorithm.

A slightly more sophisticated way of generating initial conditions for λ is discussed in [13]. Since the technique is clearly explained in the reference, it will not be described here.

Both the algorithm described above and the one referenced have been applied to the Minimax Problem with little success. Whether or not convergent solutions could be obtained seemed to depend on the initial guess of $\lambda(0)$. The author of [13] has made the same observation in his paper.

These two computational techniques were not investigated extensively. Thus there will be no further discussion of them in this report.

2.9 Quasilinearization

Another computational technique which ultimately leads to the generation of improved initial values for $\lambda(t)$ is Quasilinearization [4]. The convergence properties of this technique have been studied by Kalaba [14] who has established that the algorithm provides monotonic convergence for a restricted class of problems.

To use Quasilinearization to solve the boundary value problem described in the last section, it is necessary that all first partials of the functions M and N exist. M will usually contain terms like sat $(B'\lambda)$ and sgn $(B'\lambda)$ (see equations (2.28) to (2.31)). The following approximations have been used for the partials of these functions:

$$\frac{\partial \text{ sat } (B'\lambda)}{\partial \lambda_i} = \left\{ \begin{array}{ll} 0 & \text{if } |B'\lambda| > 1 \\ B_i & \text{if } |B'\lambda| \leq 1 \end{array} \right\} \tag{2.60}$$

where λ_i and B_i are the ith components of the vectors λ and B. The sgn function is first approximated as a sat function with slope K where K is large:

$$\text{sgn } (B'\lambda) \approx \text{sat } (KB'\lambda) \tag{2.61}$$

Then

$$\frac{\partial \text{ sgn } (B'\lambda)}{\partial \lambda_i} \approx \frac{\partial \text{ sat } (KB'\lambda)}{\partial \lambda_i} = \left\{ \begin{array}{ll} 0 & \text{if } |KB'\lambda| > 1 \\ KB_i & \text{if } |KB'\lambda| \leq 1 \end{array} \right\} \tag{2.62}$$

Just what value K should have for a particular problem is not clear. Making K too small will make the approximation to the sgn function poor while if K is too large, the partials of the sgn function will be zero except for very small values of the λ_i. This will of course, affect convergence.

In problem 2.6 (see section 2.14), K = 10000 worked successfully while in problem 2.7, the same value of K prevented the algorithm from converging. K = 10 proved to be satisfactory for problem 2.7.

With the understanding that the above approximations will be used whenever necessary, we now turn to a discussion of Quasilinearization. Let the 2n-vector

$$\begin{bmatrix} x^m(t) \\ \lambda^m(t) \end{bmatrix}$$

be a known solution of (2.56) and (2.57) which does not necessarily satisfy the boundary conditions (2.58) and (2.59). Our aim will be to find a new solution

$$\begin{bmatrix} x^{m+1}(t) \\ \lambda^{m+1}(t) \end{bmatrix}$$

which will come closer to satisfying these boundary conditions than the mth solution. To do this, M and N are first expanded about the known mth solution. Thus

$$\begin{bmatrix} \dot{x}^{m+1} \\ \dot{\lambda}^{m+1} \end{bmatrix} = \begin{bmatrix} M(x^m(t) , \lambda^m(t)) \\ N(x^m(t) , \lambda^m(t)) \end{bmatrix} + J(x^m(t) , \lambda^m(t)) \begin{bmatrix} x^{m+1} - x^m \\ \lambda^{m+1} - \lambda^m \end{bmatrix} + \begin{matrix} \text{higher} \\ \text{order} \\ \text{terms} \end{matrix} \tag{2.63}$$

where

$$J(x^m(t), \lambda^m(t)) = \begin{bmatrix} \frac{\partial M_1(x^m(t), \lambda^m(t))}{\partial x_1} & \cdots & \frac{\partial M_1(x^m(t), \lambda^m(t))}{\partial x_n} & \cdots & \frac{\partial M_1(x^m(t), \lambda^m(t))}{\partial \lambda_1} & \cdots & \frac{\partial M_1(x^m(t), \lambda^m(t))}{\partial \lambda_n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \frac{\partial M_n(x^m(t), \lambda^m(t))}{\partial x_1} & \cdots & \frac{\partial M_n(x^m(t), \lambda^m(t))}{\partial x_n} & \cdots & \frac{\partial M_n(x^m(t), \lambda^m(t))}{\partial \lambda_1} & \cdots & \frac{\partial M_n(x^m(t), \lambda^m(t))}{\partial \lambda_n} \\ \frac{\partial N_1(x^m(t), \lambda^m(t))}{\partial x_1} & \cdots & \frac{\partial N_1(x^m(t), \lambda^m(t))}{\partial x_n} & \cdots & \frac{\partial N_1(x^m(t), \lambda^m(t))}{\partial \lambda_1} & \cdots & \frac{\partial N_1(x^m(t), \lambda^m(t))}{\partial \lambda_n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \frac{\partial N_n(x^m(t), \lambda^m(t))}{\partial x_1} & \cdots & \frac{\partial N_n(x^m(t), \lambda^m(t))}{\partial x_n} & \cdots & \frac{\partial N_n(x^m(t), \lambda^m(t))}{\partial \lambda_1} & \cdots & \frac{\partial N_n(x^m(t), \lambda^m(t))}{\partial \lambda_n} \end{bmatrix}$$

Equation (2.64)

which is recognized as the Jacobian of the vector function

$$\begin{bmatrix} M(x^m(t),\ \lambda^m(t)) \\ N(x^m(t),\ \lambda^m(t)) \end{bmatrix}$$

which is a 2n-dimensional vector. Dropping higher order terms in (2.63) and rearranging

$$\begin{bmatrix} \dot{x}^{m+1} \\ \dot{\lambda}^{m+1} \end{bmatrix} = J(x^m(t),\ \lambda^m(t)) \begin{bmatrix} x^{m+1} \\ \lambda^{m+1} \end{bmatrix} + \begin{bmatrix} M(x^m(t),\ \lambda^m(t)) \\ N(x^m(t),\ \lambda^m(t)) \end{bmatrix}$$

$$- J(x^m(t),\ \lambda^m(t)) \begin{bmatrix} x^m \\ \lambda^m \end{bmatrix} \tag{2.65}$$

Since

$$\begin{bmatrix} x^m(t) \\ \lambda^m(t) \end{bmatrix}$$

are known functions of time, equation (2.65) is a <u>linear</u> time-varying differential equation. It is referred to as the quasilinearized form of (2.56) and (2.57).

Since (2.65) is linear, its solution will be of the <u>form</u>

$$\begin{bmatrix} x^{m+1}(t) \\ \lambda^{m+1}(t) \end{bmatrix} = \Phi^{m+1}(t) \begin{bmatrix} x^{m+1}(0) \\ \lambda^{m+1}(0) \end{bmatrix} + \begin{bmatrix} P_x{}^{m+1}(t) \\ P_\lambda{}^{m+1}(t) \end{bmatrix} \tag{2.66}$$

where $\Phi^{m+1}(t)$ is the fundamental matrix[1] (2nx2n) satisfying

$$\dot{\Phi}^{m+1}(t) = J(x^m(t),\ \lambda^m(t))\ \Phi^{m+1}(t) \tag{2.67}$$

with initial conditions

$$\Phi^{m+1}(0) = I \qquad \text{(identity matrix)} \tag{2.68}$$

---

1) See [15], Chapter 1.

$\begin{bmatrix} P_x^{m+1}(t) \\ P_\lambda^{m+1}(t) \end{bmatrix}$ is the particular solution (2n-vector) satisfying

$$\begin{bmatrix} \dot{P}_x^{m+1}(t) \\ \dot{P}_\lambda^{m+1}(t) \end{bmatrix} = J(x^m(t), \lambda^m(t)) \begin{bmatrix} P_x^m(t) \\ P_\lambda^m(t) \end{bmatrix} + \begin{bmatrix} M(x^m(t), \lambda^m(t)) \\ N(x^m(t), \lambda^m(t)) \end{bmatrix}$$

$$- J(x^m(t), \lambda^m(t)) \begin{bmatrix} x^m(t) \\ \lambda^m(t) \end{bmatrix} \tag{2.69}$$

with initial conditions

$$\begin{bmatrix} P_x^{m+1}(0) \\ P_\lambda^{m+1}(0) \end{bmatrix} = 0 \tag{2.70}$$

Since the mth solution

$$\begin{bmatrix} x^m(t) \\ \lambda^m(t) \end{bmatrix}$$

is known, it is possible to compute $\Phi^{m+1}(t)$ and

$$\begin{bmatrix} Px^{m+1}(t) \\ P_\lambda^{m+1}(t) \end{bmatrix}$$

using equations (2.67) to (2.70). Having computed these functions, $x^{m+1}(0)$ and $\lambda^{m+1}(T)$ are chosen so that the m+1st linearized solution in (2.65) satisfies the boundary conditions in (2.58) and (2.59).

Specifically, let

$$x^{m+1}(0) = x_o \tag{2.71}$$

and

$$\lambda^{m+1}(T) = P' x^{m+1}(T) \tag{2.72}$$

At this point it is advantageous to set P = 0. Though this is not a necessary step, it turns out to make things computationally easier. In terms of the Minimax Problem (see equation (2.19)) this means that a Lagrange problem is being considered instead of a Bolza problem. However there is no loss of generality in doing this since a Bolza problem may easily be transformed into a Lagrange problem. Thus, assuming P = 0, from (2.72)

$$\lambda^{m+1}(T) = 0 \tag{2.73}$$

Consider the partitioned form of the $\Phi^{m+1}(t)$ matrix

$$\Phi^{m+1}(t) = \begin{bmatrix} \Phi_{xx}^{m+1}(t) & \Phi_{x\lambda}^{m+1}(t) \\ \Phi_{\lambda x}^{m+1}(t) & \Phi_{\lambda\lambda}^{m+1}(t) \end{bmatrix} \tag{2.74}$$

From (2.66) and (2.74) it is clear that

$$\lambda^{m+1}(T) = \Phi_{\lambda x}^{m+1}(T)\, x^{m+1}(0) + \Phi_{\lambda\lambda}^{m+1}(T)\, \lambda^{m+1}(0) + P_{\lambda}^{m+1}(T) \tag{2.75}$$

Solving for $\lambda^{m+1}(0)$ and substituting in (2.71) and (2.73)

$$\lambda^{m+1}(0) = -\left(\Phi_{\lambda\lambda}^{m+1}(T)\right)^{-1} \left(P_{\lambda}^{m+1}(T) + \Phi_{\lambda x}^{m+1}(T)\, x_o\right) \tag{2.76}$$

Hence if (2.71) and (2.76) are used as initial conditions for (2.65), the solution (2.66) will satisfy the desired boundary conditions indicated in (2.58) and (2.59). Note that making P = 0 has greatly simplified the form of equation (2.76).

Based on the above discussion, the Quasilinearization algorithm proposed by Bellman [4] is as follows:

1. Guess some initial $\lambda^{o}(0)$ and integrate (2.56) and (2.57) to obtain a first solution $\begin{bmatrix} x^{o}(t) \\ \lambda^{o}(t) \end{bmatrix}$ for m = 0.

2. Calculate $\Phi^{m+1}(t)$ and $\begin{bmatrix} P_x^{m+1}(t) \\ P_\lambda^{m+1}(t) \end{bmatrix}$ from (2.67) to (2.70).

3. Using (2.71) and (2.76) obtain $x^{m+1}(0)$ and $\lambda^{m+1}(0)$.

4. Calculate $\begin{bmatrix} x^{m+1}(t) \\ \lambda^{m+1}(t) \end{bmatrix}$ using (2.66).

5. Let m = m+1 and go to 2.

The iterations should continue until the trajectories obtained from two successive iterations differ by less than some preassigned amount.

Note that when the m and m+1st solutions are the same, the quasilinearized differential equation in (2.65) is the same as the original differential equations in (2.56) and (2.57). Thus a solution to the quasilinearized equation is the same as a solution to the original nonlinear differential equations when convergence is achieved.

Inspection of the above algorithm indicates that $4n^2 + 2n$ differential equations must be solved during each iteration. For a third-order plant this amounts to 42 equations. The time required to compute the solutions to this many equations can be quite large.

An even more severe problem with the above algorithm is the computer memory requirement. Both $\Phi(t)$ and $\begin{bmatrix} P_x(t) \\ P_\lambda(t) \end{bmatrix}$ must be stored in memory. Thus there must be room in the computer to store at least $4n^2 + 2n$ time functions. The computer memory needed for the storage of this many time functions can be excessive, even for low order plants.

By making several changes in the algorithm described above, it is possible to reduce the number of differential equations to be solved to $2n^2 + 2n$. It is also possible to essentially eliminate the storage problem. The appropriate changes are indicated in the following discussion.

Suppose that the mth solution $\begin{bmatrix} x^m(t) \\ \lambda^m(t) \end{bmatrix}$ satisfies the original differential equation

$$\begin{bmatrix} \dot{x}^m \\ \dot{\lambda}^m \end{bmatrix} = \begin{bmatrix} M(x^m, \lambda^m) \\ N(x^m, \lambda^m) \end{bmatrix} \tag{2.77}$$

with initial conditions

$$\begin{bmatrix} x^m(0) \\ \lambda^m(0) \end{bmatrix} = \begin{bmatrix} x_o \\ \lambda^m(0) \end{bmatrix} \tag{2.78}$$

It will be helpful to define the new variables

$$\delta_x^{m+1}(t) = x^{m+1}(t) - x^m(t) \tag{2.79}$$

and

$$\delta_\lambda^{m+1}(t) = \lambda^{m+1}(t) - \lambda^m(t) \tag{2.80}$$

Using (2.79) and (2.80), the quasilinearized equation can be written as

$$\begin{bmatrix} \dot{x}^m + \dot{\delta}_x^{m+1} \\ \dot{\lambda}^m + \dot{\delta}_\lambda^{m+1} \end{bmatrix} = J(x^m(t), \lambda^m(t)) \begin{bmatrix} \delta_x^{m+1} \\ \delta_\lambda^{m+1} \end{bmatrix} + \begin{bmatrix} M(x^m, \lambda^m) \\ N(x^m, \lambda^m) \end{bmatrix} \tag{2.81}$$

From equation (2.77) it follows that

$$\begin{bmatrix} \dot{\delta}_x^{m+1} \\ \dot{\delta}_\lambda^{m+1} \end{bmatrix} = J(x^m(t), \lambda^m(t)) \begin{bmatrix} \delta_x^{m+1} \\ \delta_\lambda^{m+1} \end{bmatrix} \tag{2.82}$$

Using (2.71), (2.73), and (2.77) to (2.80)

$$\delta_x^{m+1}(0) = 0$$

and (2.83)

$$\delta_\lambda^{m+1}(T) = - \lambda^m(T)$$

The solution of (2.82) is of the form

$$\begin{bmatrix} \delta_x^{m+1}(t) \\ \delta_\lambda^{m+1}(t) \end{bmatrix} = \begin{bmatrix} \Phi_{xx}^{m+1}(t) & \Phi_{x\lambda}^{m+1}(t) \\ \Phi_{\lambda x}^{m+1}(t) & \Phi_{\lambda\lambda}^{m+1}(t) \end{bmatrix} \begin{bmatrix} \delta_x^{m+1}(0) \\ \delta_\lambda^{m+1}(0) \end{bmatrix} \tag{2.84}$$

Therefore

$$\delta_\lambda^{m+1}(T) = \Phi_{\lambda\lambda}^{m+1}(T)\, \delta_\lambda^{m+1}(0) \tag{2.85}$$

and

$$\delta_\lambda(0) = (\Phi_{\lambda\lambda}^{m+1}(T))^{-1}\, \delta_\lambda^{m+1}(T) \tag{2.86}$$

or

$$\delta_\lambda(0) = -(\Phi_{\lambda\lambda}^{m+1}(T))^{-1}\, \lambda^m(T) \tag{2.87}$$

Finally, from (2.80)

$$\lambda^{m+1}(0) = \lambda^m(0) - (\Phi_{\lambda\lambda}^{m+1}(T))^{-1}\, \lambda^m(T) \tag{2.88}$$

From (2.67), (2.68) and (2.74) it can be seen that

$$\begin{bmatrix} \dot{\Phi}_{x\lambda}^{m+1}(t) \\ \dot{\Phi}_{\lambda\lambda}^{m+1}(t) \end{bmatrix} = J(x^m(t),\, \lambda^m(t)) \begin{bmatrix} \Phi_{x\lambda}^{m+1}(t) \\ \Phi_{\lambda\lambda}^{m+1}(t) \end{bmatrix} \tag{2.89}$$

with initial conditions

$$\begin{bmatrix} \Phi_{x\lambda}^{m+1}(0) \\ \Phi_{\lambda\lambda}^{m+1}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{2.90}$$

Using the results indicated above, the following algorithm is proposed:

1. Guess some initial $\lambda^o(0)$ for $m = 0$.
2. Simultaneously solve (2.77) and (2.89) using initial conditions (2.78) and (2.90). Store only $\lambda^m(T)$ and $\Phi_{\lambda\lambda}^{m+1}(T)$.
3. Use (2.88) to obtain $\lambda^{m+1}(0)$.
4. Set $m = m+1$ and go to 2.

As in the previous algorithm, iterations should continue until the trajectories obtained from two successive iterations differ by less than some preassigned amount.

In this algorithm no time functions need be stored eliminating the large memory problem associated with the first algorithm. It is further noted that the number of differential equations to be solved in step 2 is $2n^2 + 2n$.

The second Quasilinearization algorithm described above has been applied to problems 2.1 to 2.10 (see section 2.14). Solutions were obtained for all of these problems except 2.10. For the problems with T large compared to plant settling time (problems 2.6 to 2.10), it was quite difficult to obtain solutions. Usually these problems had to be run 10 to 20 times with different initial guesses for $\lambda^o(0)$ before solutions could be obtained.

There are two factors which appear to have an important effect on convergence. The first is the sat and sgn function approximations described earlier in this section. The validity of the approximations has not been established. Just how these approximations affect convergence is a subject for further study.

The second factor thought to affect convergence is the sensitivity of $\lambda(T)$ to small changes in $\lambda(0)$. It has been found in some problems that small changes in $\lambda(0)$ can cause changes in $\lambda(T)$ which are many times larger. For larger values of T, this sensitivity problem becomes more acute. The computation of $\lambda^{m+1}(0)$ requires (among other things) the inversion of $\phi_{\lambda\lambda}^{m+1}(T)$ (see equation (2.88)). A slight error in the calculation of $\lambda^{m+1}(0)$ due to computer round-off could therefore cause a very large error in $\lambda^{m+1}(T)$. For this reason, Quasilinearization does not appear to be a good computational technique for the type of problems we have considered when T is large.

This sensitivity problem is illustrated graphically in figure 2.1. The normalized values of the components of $\lambda^m(0)$ and $\lambda^m(T)$ are plotted as functions of iteration number m for the last six iterations made in obtaining a solution to problem 2.7 (see section 2.14). All six curves are plotted to the same scale. Note that the changes in the components of $\lambda^m(T)$ are much larger than those for $\lambda^m(0)$.

2.10 The Ping Pong Algorithm

The approach used in the computational techniques discussed thus far, has been to generate improved initial conditions for λ. In doing this, equations (2.28) to (2.31) have been used to eliminate u and v from the plant equations.

An alternate approach to the problem outlined in section 2.6 is to let x and λ satisfy the desired boundary conditions all the time and then to choose u and v so that (2.27) is satisfied. Since improvement is to be made by generating better and better choices of u and v rather than $\lambda(0)$, the approach is a form of "Approximation in Policy Space" [4].[1] The technique has the advantage that the partials of sat and sgn functions are not required.[2] ,

In the algorithm described below, it is assumed that u and v are always chosen so as to satisfy the hard constraints in equation (2.18). The following is an outline of the "Ping Pong" algorithm:

---

1) Another computational scheme which is also a form of Approximation in Policy Space, is presented in Appendix D. The technique is not included here because in its present form it is not applicable to the Minimax Problem. It is felt however, that the technique will be of interest to MSFC and it is therefore included as an appendix.

2) For a further discussion of some of the advantages and disadvantages of approximating in a policy space as opposed to varying boundary conditions, see the first part of chapter 10 in [16].

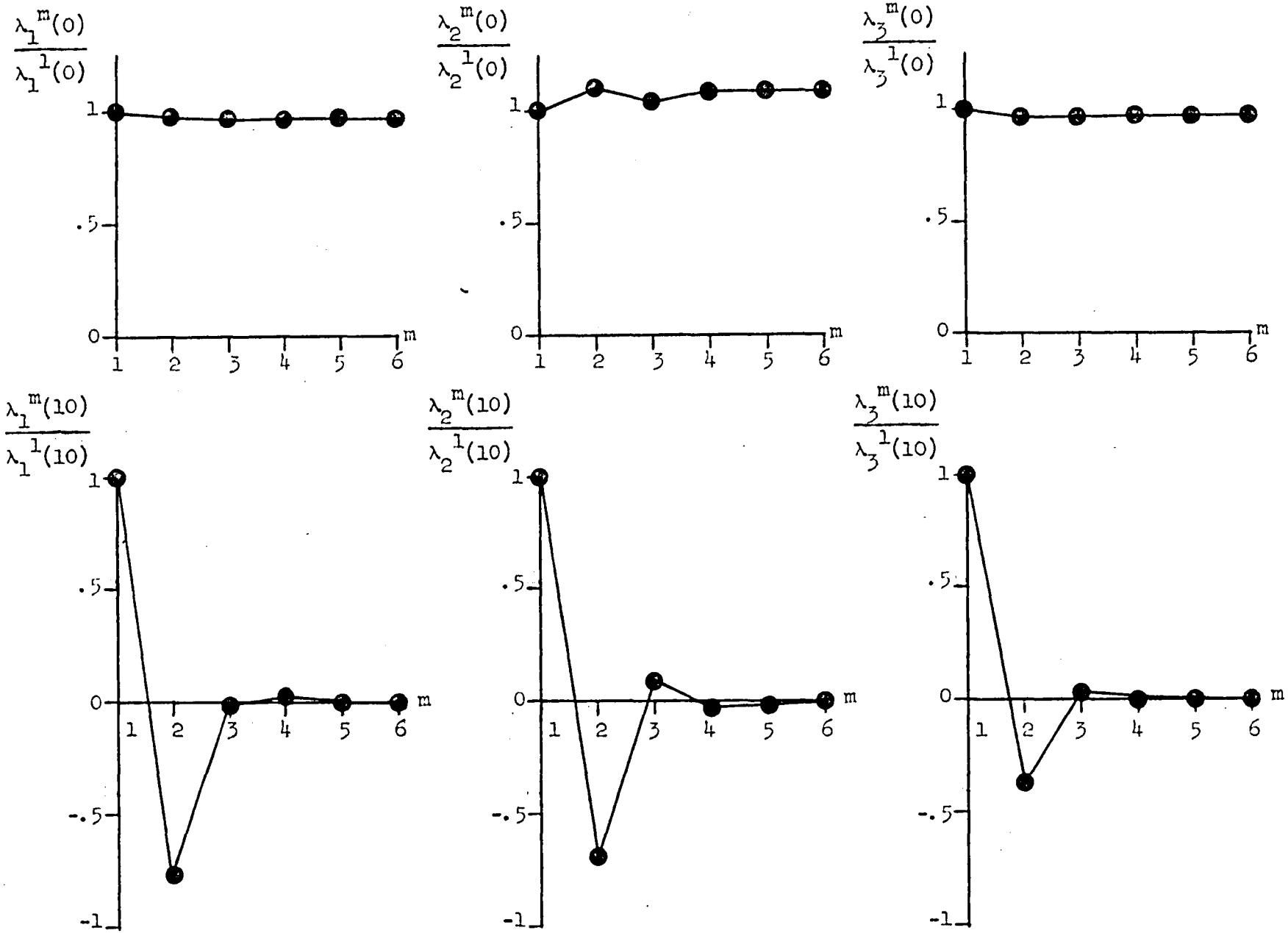


Figure 2.1

Variations in Initial and Final Values of $\lambda(t)$ as Functions of Iteration

1. Guess $u^o(t)$ and $v^o(t)$ and set $m = 0$.
2. Integrate the plant equation in (2.16) with initial condition (2.17) to obtain $x^m(t)$.
3. Set $\lambda^m(T) = P'x^m(T)$
4. Integrate the adjoint equation (2.25) backwards in time to obtain $\lambda^m(t)$.
5. Choose $u^{m+1}(t)$ and $v^{m+1}(t)$ so that
$$H(x^m(t), \lambda^m(t), u^{m+1}(t), v(t)) \leq H(x^m(t), \lambda^m(t), u^{m+1}(t), v^{m+1}(t))$$
$$\leq H(x^m(t), \lambda^m(t), u(t), v^{m+1}(t))$$
for all admissible $u(t)$ and $v(t)$.
6. Set $m = m+1$ and go to 2.

Iterations should continue until the $u(t)$ and $v(t)$ obtained on two successive iterations differ by less than some preassigned amount.

The inequalities in Step 5 will be satisfied if

$$v^{m+1}(t) = \begin{cases} V_2 & \text{if } < \lambda^m, C > > 0 \\ V_1 & \text{if } < \lambda^m, C > < 0 \end{cases} \qquad (2.91)$$

and

$$u^{m+1}(t) = \begin{cases} U_2 & \text{if } < \lambda^m, B > < 0 \\ U_1 & \text{if } < \lambda^m, B > > 0 \end{cases} \qquad (2.92)$$

or

$$u^{m+1}(t) = \begin{cases} U_2 & \text{if } U_2 < -\dfrac{< \lambda^m, B >}{a} \\ -\dfrac{< \lambda^m, B >}{a} & \text{if } U_1 \leq -\dfrac{< \lambda^m, B >}{a} \leq U_2 \\ U_1 & \text{if } U_1 > -\dfrac{< \lambda^m, B >}{a} \end{cases} \qquad (2.93)$$

This algorithm has been applied to some of the problems in section 2.14. It worked well for some (problems 2.1 to 2.4) and failed for others (problems 2.6 to 2.8).

One reason why the algorithm doesn't work well in all sample problems is because the changes in u and v between iterations are too large. To understand why this is so, it will be helpful to first investigate steepest descent techniques. Thus further discussion of this algorithm is deferred until a later section.

2.11 The Gradient Technique

In the preceding sections, the Minimax Problem was formulated and the approach taken to solve the problem was to use the Minimax Principle. It was shown that such an approach required the solution of a two-point boundary value problem.

A somewhat different approach can be taken to solve the Minimax Problem. This approach is basically a Gradient Technique. Besides being a good computational scheme, the Gradient Technique gives further insight into the Minimax Problem.

The idea of moving in the direction of a gradient or in the direction of steepest descent (ascent) is not new. Amongst others, Bryson [6], [8], Kelley [9], Ho and Brentani [7], and Dreyfus [5] have done considerable work with this technique. However, to our knowledge, none of these researchers has applied the technique to a Minimax Problem.

Initially the technique will be developed for a discrete system. There are three reasons for doing this:

1. The derivation of the necessary equations is greatly facilitated by using a discrete system.
2. The discrete problem readily lends itself to geometric interpretation, making it easy to visualize what is happening.
3. Even continuous problems must somehow be discretized since the Gradient Technique is normally executed on a digital computer. After developing the necessary equations, it will be possible to alter the technique for continuous systems. In doing this, it will become apparent that the Gradient Technique is quite similar to the Ping Pong algorithm.

The discussion starts with a restatement of the Minimax Problem for a discrete version of the problem outlined at the beginning of section 2.6:

1. The plant to be controlled is described by a set of n recursion equations of the form

$$\hat{x}_{i+1} = \hat{A}\hat{x}_i + \hat{B}\hat{u}_i + \hat{C}\hat{v}_i \qquad (2.94)$$

where $\hat{x}_i$ is an n-vector describing the state of the plant

$\hat{A}$ is a constant n x n matrix

$\hat{B}$ and $\hat{C}$ are constant n-vectors

$\hat{u}_i$ and $\hat{v}_i$ are scalars, $\hat{u}_i$ being the control input and $\hat{v}_i$ being the disturbance input.

The $\hat{}$ symbol is used to distinguish the variables used here from those for the continuous problem.

2. The initial conditions

$$\hat{x}_1 = x_o \qquad (2.95)$$

are assumed to be known.

3. The disturbance $\hat{v}_i$ is restricted to lie between fixed bounds. A similar restriction is placed on the control $\hat{u}_i$ in some problems.

$$V_1 \leq \hat{v}_i \leq V_2$$
$$U_1 \leq \hat{u}_i \leq U_2 \tag{2.96}$$

4. The index i goes from 1 to N+1 where N is a fixed number.

5. The values of the state variables are free for i = N+1.

6. The performance index is of the form

$$\hat{J}(\hat{u}, \hat{v}) = \frac{1}{2} \sum_{i=1}^{N} [\hat{x}_i' \hat{Q} \hat{x}_i + \hat{a}\hat{u}_i^2 + 2\hat{b}'\hat{x}_i] + \frac{1}{2} \hat{x}_{N+1}' \hat{P} \hat{x}_{N+1} \tag{2.97}$$

where $\hat{Q}$ and $\hat{P}$ are constant, n x n, positive-semidefinite matrices

$\hat{b}$ is a constant n-vector

$\hat{a}$ is a non-negative scalar

$\hat{u}$ and $\hat{v}$ are each N-vectors whose ith components are $\hat{u}_i$ and $\hat{v}_i$ respectively.

7. It is desired to find $\hat{u}^*$ and $\hat{v}^*$ such that

$$\hat{J}(\hat{u}^*, \hat{v}) \leq \hat{J}(\hat{u}^*, \hat{v}^*) \leq \hat{J}(\hat{u}, \hat{v}^*) \tag{2.98}$$

for all admissible $\hat{u}$ and $\hat{v}$.

If (2.94) is substituted into (2.97) it is clear that $\hat{J}(\hat{u}, \hat{v})$ becomes a function of only $\hat{u}$, $\hat{v}$ and the fixed vector x_o. One may visualize a 2N-dimensional space with the $\hat{u}_i$ and $\hat{v}_i$ as the coordinates of the space. Since $\hat{J}(\hat{u}, \hat{v})$ is a quadratic function of the $\hat{u}_i$ and $\hat{v}_i$, the loci of constant $\hat{J}(\hat{u}, \hat{v})$ will be closed hyperellipsoids. For example, for N = 1, the loci might look like those in figure 2.2. The gradient of $\hat{J}(\hat{u}, \hat{v})$, $\nabla\hat{J}$, with respect to $\hat{u}$ and $\hat{v}$, is indicated at a point on one of the level curves. Note that it points in the direction of steepest ascent. For this simple example, the components of $\nabla\hat{J}$ in the directions of the coordinates $\hat{u}_1$ and $\hat{v}_1$ are $\nabla_u\hat{J}$ and $\nabla_v\hat{J}$. In general,

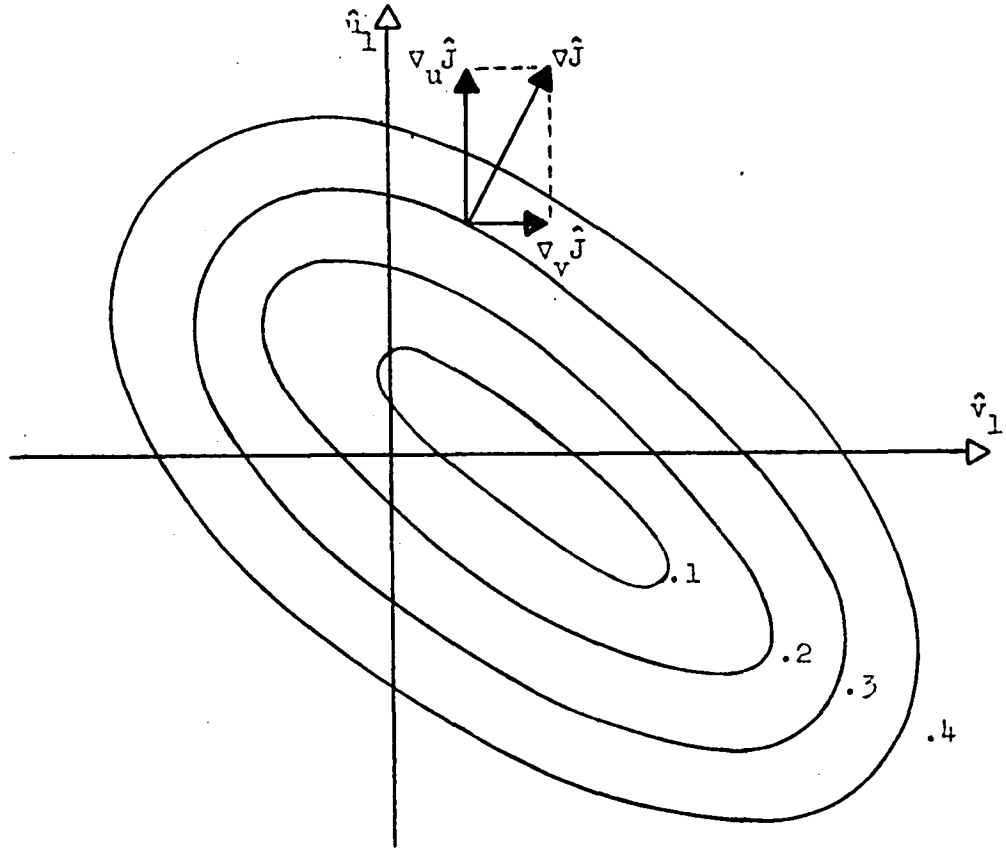


Figure 2.2

Level Curves for the Performance Index in Equation (2.98) with N = 1.

for a 2N-dimensional space

$$\nabla \hat{J} = \begin{bmatrix} \nabla_u \hat{J} \\ \nabla_v \hat{J} \end{bmatrix} \tag{2.99}$$

where $\nabla_u \hat{J}$ and $\nabla_v \hat{J}$ are each N-dimensional vectors. It is clear that $\nabla_u \hat{J}$ and $\nabla_v \hat{J}$ point in the directions of steepest ascent for $\hat{u}$ and $\hat{v}$ respectively. Since it is desired to minimize $\hat{J}(\hat{u}, \hat{v})$ with respect to $\hat{u}$, the motion of $\hat{u}$ should be in the $-\nabla_u \hat{J}$ direction. Using similar reasoning, to maximize with respect to $\hat{v}$, the motion of $\hat{v}$ should be in the $\nabla_v \hat{J}$ direction.

Two things need to be done. First, $\nabla_u \hat{J}$ and $\nabla_v \hat{J}$ must be computed. Second, it has to be determined how far to move in each direction during each iteration. The following discussion is concerned with the first of these two tasks.

It is clear that the ith component of $\nabla_u \hat{J}$ is $\frac{\partial \hat{J}(\hat{u},\hat{v})}{\partial \hat{u}_i}$. Similarly the ith component of $\nabla_v \hat{J}$ is $\frac{\partial \hat{J}(\hat{u},\hat{v})}{\partial \hat{v}_i}$. Our immediate effort then, is to compute these partials for i = 1 to N.

Define the function

$$F_i \triangleq \frac{1}{2} \sum_{j=i}^{N} [\hat{x}_j' \hat{Q} \hat{x}_j + \hat{a} \hat{u}_j^2 + 2\hat{b}' \hat{x}_j] + \frac{1}{2} \hat{x}_{N+1}' \hat{P} \hat{x}_{N+1} \qquad (2.100)$$

From this definition it is clear that

$$F_i = F_{i+1} + \frac{1}{2} \hat{x}_i' \hat{Q} \hat{x}_i + \frac{1}{2} \hat{a} \hat{u}_i^2 + \hat{b}' \hat{x}_i \quad (i = 1 \text{ to } N) \qquad (2.101)$$

and

$$F_{N+1} = \frac{1}{2} \hat{x}_{N+1}' \hat{P} \hat{x}_{N+1} \qquad (2.102)$$

and

$$F_1 = \hat{J}(\hat{u}, \hat{v}) \qquad (2.103)$$

From equation (2.101) note that

$$\frac{\partial F_i}{\partial \hat{u}_i} = \frac{\partial F_j}{\partial \hat{u}_i} \quad (i = 1 \text{ to } N, \; j = 1 \text{ to } i) \qquad (2.104)$$

In particular, for j = 1

$$\frac{\partial F_i}{\partial \hat{u}_i} = \frac{\partial F_1}{\partial \hat{u}_i} = \frac{\partial \hat{J}(\hat{u}, \hat{v})}{\partial \hat{u}_i} \quad (i = 1 \text{ to } N) \qquad (2.105)$$

Thus to find the ith component of $\nabla_u \hat{J}$, it will suffice to find $\frac{\partial F_i}{\partial \hat{u}_i}$. Similarly, the ith component of $\nabla_v \hat{J}$ is $\frac{\partial F_i}{\partial \hat{v}_i}$. Using (2.101)

$$\frac{\partial F_i}{\partial \hat{u}_i} = < \nabla_{\hat{x}_{i+1}} F_{i+1}, \frac{\partial \hat{x}_{i+1}}{\partial \hat{u}_i} > + \hat{a} \hat{u}_i \quad (i = 1 \text{ to } N) \qquad (2.106)$$

Using (2.101) again

$$\nabla_{\hat{x}_i} F_i = \left(\frac{\partial \hat{x}_{i+1}}{\partial \hat{x}_i} \right)' \nabla_{\hat{x}_{i+1}} F_{i+1} + \hat{Q}' \hat{x}_i + \hat{b} \quad (i = 1 \text{ to } N) \tag{2.107}$$

where $\left(\frac{\partial \hat{x}_{i+1}}{\partial \hat{x}_i} \right)$ is the Jacobian of the vector $\hat{x}_{i+1}$ with respect to the vector $\hat{x}_i$. In terms of equation (2.94)

$$\left(\frac{\partial \hat{x}_{i+1}}{\partial \hat{x}_i} \right) = \hat{A} \qquad (i = 1 \text{ to } N) \tag{2.108}$$

and

$$\frac{\partial \hat{x}_{i+1}}{\partial \hat{u}_i} = \hat{B} \qquad (i = 1 \text{ to } N) \tag{2.109}$$

For reasons to become apparent shortly, define

$$\hat{\lambda}_i \triangleq \nabla_{\hat{x}_i} F_i \qquad (i = 1 \text{ to } N+1) \tag{2.110}$$

Using the results in the above equations, it follows that

$$\frac{\partial \hat{J}(\hat{u}, \hat{v})}{\partial \hat{u}_i} = < \hat{B}, \hat{\lambda}_{i+1} > + \hat{a}\, \hat{u}_i \qquad (i = 1 \text{ to } N) \tag{2.111}$$

The corresponding expression for $\hat{v}$ is

$$\frac{\partial \hat{J}(\hat{u}, \hat{v})}{\partial \hat{v}_i} = < \hat{C}, \hat{\lambda}_{i+1} > \qquad (i = 1 \text{ to } N) \tag{2.112}$$

From equation (2.107)

$$\hat{\lambda}_i = \hat{A}' \hat{\lambda}_{i+1} + \hat{Q}' \hat{x}_i + \hat{b} \qquad (i = 1 \text{ to } N) \tag{2.113}$$

Using (2.102)

$$\hat{\lambda}_{N+1} = \hat{P}' \hat{x}_{N+1} \tag{2.114}$$

Thus by using equations (2.113) and (2.114), all of the $\hat{\lambda}_i$'s may be calculated. These in turn may be used to evaluate the partials in (2.111) and (2.112).

The way in which these gradients are used in searching for Minimax solutions is as follows. Suppose that after the mth iteration, the control

and disturbance vectors are $\hat{u}^m$ and $\hat{v}^m$. Corresponding to these vectors are the mth trajectory $\hat{x}_i^m$ and the performance index $\hat{J}(\hat{u}^m, \hat{v}^m)$. In view of equation (2.98), it seems reasonable to change $\hat{u}^m$ in a way such that the performance index will probably be decreased. Similarly, $\hat{v}^m$ should be altered so as to tend to increase the performance index. Specifically, let

$$\hat{u}^{m+1} = - K_u^m \nabla_u \hat{J}^m + \hat{u}^m \tag{2.115}$$

and

$$\hat{v}^{m+1} = K_v^m \nabla_v \hat{J}^m + \hat{v}^m \tag{2.116}$$

where $\nabla_u \hat{J}^m$ and $\nabla_v \hat{J}^m$ are determined from equations (2.111) to (2.114) using $\hat{x}_i^m$. The positive scalars K_u^m and K_v^m, which will be referred to as step sizes, need to be determined. This is the subject of the next section.

2.12 Step Size

There has been a considerable amount of work done on the step size problem [7], [8], [9], [16]. Reference is made to the well-known method proposed by Bryson [8]. In his method, a desired or expected change in performance index is chosen for each iteration. For example, in a maximization problem, the expected change in the performance index is a positive number. From this, an approximate value of step size is determined and used. If the resulting change in the performance index differs greatly from the assumed value, the assumed change is reduced and the step size recomputed. This is continued until a step size is found which is small enough so that the assumed and actual changes in performance index are in approximate agreement.

The problem in applying this method to the Minimax Problem is the choosing of the expected change in the performance which could be positive (due to a change in v), negative (due to a change in u) or either positive or negative (due to changes in both v and u). For this reason, Bryson's

method has not been used in the determination of step sizes for the Minimax Problem.

The approach which follows avoids the difficulties associated with Bryson's method. However, at present it is restricted to linear systems with quadratic performance indices.

Since the $\hat{u}_i$ and $\hat{v}_i$ are bounded as indicated in (2.96), K_u^m and K_v^m must be chosen so that these constraints are not violated. Because of this, it is convenient to use equations slightly different than those in (2.115) and (2.116). The alternate equations are

$$\hat{u}^{m+1} = K_u^m \delta_u^m + \hat{u}^m \tag{2.117}$$

and

$$\hat{v}^{m+1} = K_v^m \delta_v^m + \hat{v}^m \tag{2.118}$$

where the ith components of δ_y^m and δ_u^m (both are N-vectors) are

$$\delta_{v_i}^m = \left\{ \begin{array}{ll} V_2 - \hat{v}_i^m & \text{if } \dfrac{\partial \hat{J}(\hat{u}^m, \hat{v}^m)}{\partial \hat{v}_i} \geq 0 \\ V_1 - \hat{v}_i^m & \text{if } \dfrac{\partial \hat{J}(\hat{u}^m, \hat{v}^m)}{\partial \hat{v}_i} < 0 \end{array} \right\} \tag{2.119}$$

and

$$\delta_{u_i}^m = \left\{ \begin{array}{lll} U_2 - \hat{u}_i^m & \text{if} & -\dfrac{\partial \hat{J}(\hat{u}^m, \hat{v}^m)}{\partial \hat{u}_i} > U_2 \\ -\dfrac{\partial \hat{J}(\hat{u}^m, \hat{v}^m)}{\partial \hat{u}_i} & \text{if} & U_1 \leq -\dfrac{\partial \hat{J}(\hat{u}^m, \hat{v}^m)}{\partial \hat{u}_i} \leq U_2 \\ U_1 - \hat{u}_i^m & \text{if} & -\dfrac{\partial \hat{J}(\hat{u}^m, \hat{v}^m)}{\partial \hat{u}_i} < U_1 \end{array} \right\} \tag{2.120}$$

If both K_u^m and K_v^m are restricted to be

$$\left. \begin{array}{l} 0 \leq K_u \leq 1 \\ 0 \leq K_v \leq 1 \end{array} \right\} \tag{2.121}$$

it is clear that the hard constraints on the control and disturbance in (2.96) will be satisfied. The form of $\delta_{v_i}^m$ is influenced by the fact that

the disturbance is bang-bang (see section 2.6). When $\hat{a} = 0$, implying bang-bang control, $\delta_{u_i}{}^m$ could be changed to a form similar to $\delta_{v_i}{}^m$. Note also that the vectors $\delta_u{}^m$ and $\delta_v{}^m$ do point in the direction of decreasing and increasing performance index respectively, though these are not the steepest descent and ascent directions.

At the end of the mth iteration, $\delta_u{}^m$ and $\delta_v{}^m$ can be calculated from (2.119) and (2.120). After these vectors are determined, the step sizes $K_u{}^m$ and $K_v{}^m$ can be evaluated. To do this, it will be convenient to define

$$\Delta\hat{J}^m(L_u, L_v) \triangleq \hat{J}((\hat{u}^m + L_u \delta_u), (\hat{v}^m + L_v \delta_v)) - \hat{J}(\hat{u}^m, \hat{v}^m) \qquad (2.122)$$

where L_u and L_v are arbitrary scalars. Note that $\Delta\hat{J}^m(K_u{}^m, K_v{}^m)$ is the change in performance index between the mth and m+1st iterations.

Since the performance index is a quadratic function of $\hat{u}$ and $\hat{v}$, it is clear that $\Delta\hat{J}^m(L_u, L_v)$ is of the <u>form</u>

$$\Delta\hat{J}^m(L_u, L_v) = -\alpha_u{}^m L_u + \alpha_v{}^m L_v + \alpha_{uv}{}^m L_u L_v + \alpha_{uu}{}^m L_u{}^2 + \alpha_{vv}{}^m L_v{}^2 \qquad (2.123)$$

where
$$\left.\begin{aligned} \alpha_u{}^m &= -\langle \nabla_u \hat{J}^m, \delta_u{}^m \rangle \\ \alpha_v{}^m &= \langle \nabla_v J^m, \delta_v{}^m \rangle \end{aligned}\right\} \qquad (2.124)$$

and $\alpha_{uv}{}^m$, $\alpha_{uu}{}^m$ and $\alpha_{vv}{}^m$ are unknown scalars. It can be seen that

$$\alpha_u{}^m \geq 0 \ , \ \alpha_v{}^m \geq 0, \ \alpha_{uu}{}^m \geq 0, \ \alpha_{vv}{}^m \geq 0 \qquad (2.125)$$

while $\alpha_{uv}{}^m$ may be either positive or negative. If needed, $\alpha_{uv}{}^m$, $\alpha_{uu}{}^m$ and $\alpha_{vv}{}^m$ can be determined from

$$\alpha_{uu}{}^m = \Delta\hat{J}^m(1, 0) + \alpha_u{}^m \qquad (2.126)$$

$$\alpha_{vv}{}^m = \Delta\hat{J}^m(0, 1) - \alpha_v{}^m \qquad (2.127)$$

$$\alpha_{uv}{}^m = \Delta\hat{J}^m(1, 1) + \alpha_u - \alpha_{uu} - \alpha_v - \alpha_{vv} \qquad (2.128)$$

Note that the trajectory equation (2.94) must be solved three times, once for α_{uu}^{m}, once for α_{vv}^{m}, and once for α_{uv}^{m}.

There are several paths one can take in proceeding from this point. The two most promising of these will be presented.

In the first approach, minimization and maximization are performed on alternate iterations. That is, on the mth iteration $\Delta\hat{J}^{m}(K_u^{m}, 0)$ is minimized with respect to K_u^{m} and $K_v^{m} = 0$. Then on the m+1st iteration, $\Delta\hat{J}^{m+1}(0, K_v^{m+1})$ is maximized with respect to K_v^{m+1} and $K_u^{m+1} = 0$. On the m+2nd iteration, minimization is again performed and so on.

By taking the partial of $\Delta\hat{J}(K_u^{m}, 0)$ with respect to K_u^{m}, the optimal K_u^{m} can be obtained. With consideration to the restriction in (2.121)

$$K_u^{m} = \begin{cases} \dfrac{\alpha_u^{m}}{2\alpha_{uu}^{m}} & \text{if } \dfrac{\alpha_u^{m}}{2\alpha_{uu}^{m}} \leq 1 \\ 1 & \text{if } \dfrac{\alpha_u^{m}}{2\alpha_{uu}^{m}} > 1 \end{cases} \qquad (2.129)$$

Inspection of (2.121), (2.122) and (2.125) leads to

$$K_v^{m+1} = 1 \qquad (2.130)$$

as the value of K_v^{m+1} which maximizes $\Delta\hat{J}^{m+1}(0, K_v^{m+1})$.

With the above ideas in mind, the first Gradient Technique can be outlined as follows:

1. Guess $\hat{u}^{o}$ and $\hat{v}^{o}$ subject to (2.96) and set m = 0.
2. Calculate $\hat{x}_i^{m}$ and $\hat{J}(\hat{u}^{m}, \hat{v}^{m})$ using (2.94), (2.95) and (2.97).
3. Calculate $\hat{\lambda}_i^{m}$, δ_u^{m} and α_u^{m} using (2.111), (2.113), (2.114), (2.120), (2.124).
4. Calculate $\Delta\hat{J}^{m}(1, 0)$ using (2.94), (2.95), (2.96) and (2.122).

5. Calculate $\alpha_{uu}^{\ m}$ using (2.126).
6. Calculate $K_u^{\ m}$ using (2.129) and set $K_v^{\ m} = 0$.
7. Calculate $\hat{u}^{m+1}$ and $\hat{v}^{m+1}$ using (2.117) and (2.118).
8. Calculate $\hat{x}_i^{\ m+1}$ and $\hat{J}(\hat{u}^{m+1}, \hat{v}^{m+1})$ using (2.94), (2.95), and (2.97).
9. Calculate $\hat{\lambda}_i^{m+1}$ and $\delta_v^{\ m+1}$ using (2.112), (2.113), (2.114) and (2.119).
10. Set $K_v^{\ m+1} = 1$ and $K_u^{\ m+1} = 0$.
11. Calculate $\hat{u}^{m+2}$ and $\hat{v}^{m+2}$ using (2.117) and (2.118).
12. Set m = m+2 and go to 2.

When an improvement can no longer be made (i.e., when $\Delta\hat{J}^m(K_u^{\ m}, 0) = 0$ and $\Delta\hat{J}^{m+1}(0, K_v^{\ m+1}) = 0$) it is clear that (2.98) will be satisfied locally. Thus, iterations should continue until both $\Delta\hat{J}^m(K_u^{\ m}, 0)$ and $\Delta\hat{J}^{m+1}(0, K_v^{\ m+1})$ are less than some preassigned small number.

In the above algorithm, minimization and maximization are performed on alternate iterations. Another approach is to try to minimize and maximize $\Delta\hat{J}^m(K_u^{\ m}, K_v^{\ m})$ at the same time. From (2.121) and (2.122) the optimal step sizes are

$$K_u^{\ m} = \begin{cases} 1 & \text{if} \quad \dfrac{\alpha_u^{\ m} - \alpha_{uv}^{\ m} K_v^{\ m}}{2\alpha_{uu}} > 1 \\ \dfrac{\alpha_u^{\ m} - \alpha_{uv}^{\ m} K_v^{\ m}}{2\alpha_{uu}} & \text{if} \quad 0 \le \dfrac{\alpha_u^{\ m} - \alpha_{uv}^{\ m} K_v^{\ m}}{2\alpha_{uu}} \le 1 \\ 0 & \text{if} \quad \dfrac{\alpha_u^{\ m} - \alpha_{uv}^{\ m} K_v^{\ m}}{2\alpha_{uu}} < 0 \end{cases} \qquad (2.131)$$

and

$$K_v^{\ m} = \begin{cases} 1 & \text{if} \quad \alpha_v^{\ m} + \alpha_{uv}^{\ m} K_u^{\ m} + \alpha_{vv}^{\ m} \ge 0 \\ 0 & \text{if} \quad \alpha_v^{\ m} + \alpha_{uv}^{\ m} K_u^{\ m} + \alpha_{vv}^{\ m} < 0 \end{cases} \qquad (2.132)$$

If $\alpha_{uv}^{\ m}$ is a large negative number (which has often been the case in the problems worked) the solution of these simultaneous equations does not exist.

Because of this, the following scheme is used: First (2.131) is solved for K_u^m using $K_v^m = 1$. This K_u^m is then used with (2.132) to obtain a solution for K_v^m. If this solution agrees with the assumed value of K_v^m used in obtaining K_u^m (i.e., $K_v^m = 1$), then these are the step sizes that are used. If the solution of (2.132) is $K_v^m = 0$, then (2.131) is resolved with $K_v^m = 0$. The resulting $K_u^{m'}$ along with this value of K_v^m are used as the step sizes. Thus, if on the mth iteration, (2.131) and (2.132) cannot be simultaneously satisfied, minimization is performed with no change in disturbance. The algorithm is as follows:

1. Guess $\hat{u}^o$ and $\hat{v}^o$ subject to (2.96) and set m = 0.
2. Calculate $\hat{x}_i^m$ and $\hat{J}(\hat{u}^m, \hat{v}^m)$ using (2.94), (2.95) and (2.97).
3. Calculate $\hat{\lambda}_i^m$, δ_u^m, δ_v^m, α_u^m and α_v^m using (2.111) to (2.114), (2.119), (2.120), and (2.124).
4. Calculate $\Delta\hat{J}^m(1, 0)$ using (2.94), (2.95), (2.97) and (2.122).
5. Calculate $\Delta\hat{J}^m(0, 1)$ using (2.94), (2.95), (2.97) and (2.122).
6. Calculate $\Delta\hat{J}^m(1, 1)$ using (2.94), (2.95), (2.97) and (2.122).
7. Calculate α_{uu}^m, α_{uv}^m, α_{vv}^m using (2.126) to (2.128).
8. Calculate K_u^m with $K_v^m = 0$ using (2.131).
9. Calculate K_v^m using (2.132).
10. If $K_v^m = 1$ go to 12.
    If $K_v^m = 0$ continue.
11. Calculate K_u^m using (2.131).
12. Calculate $\hat{u}^{m+1}$ and $\hat{v}^{m+1}$ using (2.117) and (2.118).
13. Set m = m+1 and go to 2.

As with the last algorithm, when improvement can no longer be made equation (2.98) will be satisfied locally. Iterations should continue until $\Delta\hat{J}(K_u^m, K_v^m)$ is sufficiently small.

Both of these algorithms have been applied to problems 2.3, 2.7, 2.8, 2.10, 2.11 and 2.12. The results obtained in 2.3, 2.7 and 2.8 are essentially the same as those obtained via the Minimax Principle. Both algorithms also worked on problems 2.11 and 2.12. No solution could be obtained to problem 2.10. It is believed that there may be no solution to this problem.

In the various problems solved, the second algorithm usually converged more rapidly than the first. However, there are more computations required for each iteration with the second method. Which algorithm will prove to be more efficient is a subject for further study.

Both algorithms converge quite slowly when compared to Quasilinearization. However convergence itself does not seem to depend upon initial guesses. For this reason, the Gradient Technique is considered to be the best computational scheme that we have applied to the Minimax Problem.

2.13 The Gradient Technique for Continuous Systems

In the last two sections, a Gradient Technique was proposed for the solution to the Minimax Problem for discrete systems. In this section the technique will be extended to continuous systems. In doing this it will become obvious that the Gradient Technique is almost the same as the Ping Pong algorithm used with the Minimax Principle.

The development starts with a discretized version of the continuous problem outlined in section 2.6. First the time interval Δ is defined as

$$\Delta = \frac{T}{N} \quad (T \text{ is Fixed}) \tag{2.133}$$

The follwoing notation is used:

$$\left.\begin{array}{ll} x(t) = \hat{x}_i & x(t+\Delta) = \hat{x}_{i+1} \\ u(t) = \hat{u}_i & u(t+\Delta) = \hat{u}_{i+1} \\ v(t) = \hat{v}_i & v(t+\Delta) = \hat{v}_{i+1} \end{array}\right\} \tag{2.134}$$

The initial and terminal conditions are

$$\begin{aligned} x(0) &= \hat{x}_1 \\ x(T) &= \hat{x}_{N+1} \end{aligned} \tag{2.135}$$

For sufficiently small Δ, equation (2.16) may be approximated as

$$\hat{x}_{i+1} = \hat{x}_i + \Delta(A\hat{x}_i + B\hat{u}_i + C\hat{v}_i) \qquad (i = 1 \text{ to } N) \tag{2.136}$$

where from (2.17) and (2.135)

$$\hat{x}_1 = x(0) = x_o \tag{2.137}$$

are the initial conditions. A discretized version of the performance index in (2.19) is

$$\hat{J}(\hat{u}, \hat{v}) = \frac{1}{2} \sum_{i=1}^{N} [\hat{x}_i' Q\hat{x}_i + a\, \hat{u}_i^2 + 2b'\hat{x}_i] \Delta + \frac{1}{2} \hat{x}_{N+1}' P\hat{x}_{N+1} \tag{2.138}$$

Note that as $\Delta \to 0$, for fixed T, the discretized performance index becomes the same as the performance index in (2.19).

Inspection of (2.94), (2.97), (2.136) and (2.138) leads to

$$\left.\begin{aligned} \hat{Q} &= \Delta Q \\ \hat{a} &= \Delta a \\ \hat{b} &= \Delta b \\ \hat{P} &= P \\ \hat{A} &= \Delta A + I \text{ (I is an n x n identity matrix)} \\ \hat{B} &= \Delta B \\ \hat{C} &= \Delta C \end{aligned}\right\} \tag{2.139}$$

It is clear that (2.113) can be written as

$$\hat{\lambda}_i = (I + \Delta A')\, \hat{\lambda}_{i+1} + \Delta Q'\hat{x}_i + \Delta b \qquad (i = 1 \text{ to } N) \tag{2.140}$$

where

$$\hat{\lambda}_{N+1} = P'\hat{x}_{N+1} \tag{2.141}$$

Equations (2.111) and (2.112) become

$$\frac{\partial \hat{J}(\hat{u}, \hat{v})}{\partial \hat{u}_i} = \Delta\, [< B, \hat{\lambda}_{i+1} > + a\, \hat{u}_i] \tag{2.142}$$

and

$$\frac{\partial \hat{J}(\hat{u}, \hat{v})}{\partial \hat{v}_i} = \Delta\, [< C,\ \hat{\lambda}_{i+1} >] \tag{2.143}$$

$\delta_{u_i}^{\ m}$ and $\delta_{v_i}^{\ m}$ will be defined slightly differently than in equations (2.119) and (2.120).

$$\delta_{u_i}^{\ m} = \left\{ \begin{array}{ll} U_2 - \hat{u}_i^{\ m} & \text{if } -[< B, \hat{\lambda}_{i+1}^m > + a\, \hat{u}_i^{\ m}] > U_2 \\ -[< B, \hat{\lambda}_{i+1}^m > + a\, \hat{u}_i^{\ m}] & \text{if } U_1 \leq -[< B, \hat{\lambda}_{i+1}^m > + a\hat{u}_i^{\ m}] \leq U_2 \\ U_1 - \hat{u}_i^{\ m} & \text{if } -[< B, \hat{\lambda}_{i+1}^m > + a\, \hat{u}_i^{\ m}] < U_1 \end{array} \right\} \tag{2.144}$$

and

$$\delta_{v_i}^{\ m} = \left\{ \begin{array}{ll} V_2 - \hat{v}_i^{\ m} & \text{if } < C, \hat{\lambda}_{i+1}^m > \geq 0 \\ V_1 - \hat{v}_i^{\ m} & \text{if } < C, \hat{\lambda}_{i+1}^m > < 0 \end{array} \right\} \tag{2.145}$$

The reason for the altered definitions is clear. From (2.142) and (2.143) it can be seen that both $\dfrac{\partial \hat{J}^m(\hat{u}_i^{\ m}, \hat{v}_i^{\ m})}{\partial \hat{u}_i}$ and $\dfrac{\partial \hat{J}^m(\hat{u}_i^{\ m}, \hat{v}_i^{\ m})}{\partial \hat{v}_i}$ go to zero as $\Delta \to 0$. The important thing is that $\delta_u^{\ m}$ and $\delta_v^{\ m}$ still point in directions of decreasing and increasing performance index though not necessarily in the steepest descent and ascent directions.

In terms of the relations in (2.139), $\alpha_u^{\ m}$ and $\alpha_v^{\ m}$ in (2.124) become

$$\alpha_u^{\ m} = - < \nabla_u \hat{J}^m, \delta_u^{\ m} > = \sum_{i=1}^{N} [< B, \hat{\lambda}_{i+1}^m + a\, \hat{u}_i^{\ m} >] \delta_{u_i}^{\ m} \Delta \tag{2.146}$$

and

$$\alpha_v^{\ m} = < \nabla_v \hat{J}^m, \delta_v^{\ m} > = \sum_{i=1}^{N} [< C, \hat{\lambda}_{i+1}^m >] \delta_{v_i}^{\ m} \Delta \tag{2.147}$$

Now that equations (2.140) to (2.147) have been developed for the discretized version of the problem in section 2.6, we let $\Delta \to 0$ and obtain the equations for the Gradient Technique for a continuous system. As $\Delta \to 0$ it is clear that (2.140) becomes

$$\dot{\lambda} = - A'\lambda - Q'x - b \tag{2.148}$$

where

$$\lambda(T) = P'x(T) \tag{2.149}$$

It is interesting to note that these two equations are the same as (2.25) and (2.26) (assuming $\lambda_o = 1$). Thus it can be concluded that the λ's in (2.148) are the same as the λ's used in the Minimax Principle. The continuous versions of (2.144) to (2.147) are

$$\delta_u{}^m(t) = \begin{cases} U_2 - u^m(t) & \text{if} \quad -[< B, \lambda^m(t)> + au^m(t)] > U_2 \\ -[< B, {}^m(t)> + au^m(t)] & \text{if} \quad U_1 \le -[< B, \lambda^m(t)> + au^m(t)] \le U_2 \\ U_1 - u^m(t) & \text{if} \quad -[< B, \lambda^m(t)> + au^m(t)] < U_1 \end{cases} \tag{2.150}$$

$$\delta_v{}^m(t) = \begin{cases} V_2 - v^m(t) & \text{if} \quad < C, \lambda^m(t) > \ge 0 \\ V_1 - v^m(t) & \text{if} \quad < C, \lambda^m(t) > < 0 \end{cases} \tag{2.151}$$

$$\alpha_u{}^m = \int_0^T [< B, \lambda^m(t) > + au^m(t)]\, \delta_u{}^m(t)\, dt \tag{2.152}$$

$$\alpha_v{}^m = \int_0^T [< C, \lambda^m(t) >]\, \delta_v{}^m(t)\, dt \tag{2.153}$$

$\alpha_{uv}{}^m$, $\alpha_{uu}{}^m$, $\alpha_{vv}{}^m$, $K_u{}^m$ and $K_v{}^m$ can be determined in exactly the same way as for the discrete problem. The algorithms for the continuous problem are the same as those discussed in section 2.12 with the appropriate modifications indicated by (2.148) to (2.153).

The Gradient Technique for continuous systems described in this section, has not been applied to any problems. It has been included in the report mainly because it leads to a clearer understanding of the Ping Pong algorithm discussed earlier.

It should be clear at this point that the only difference between the Gradient Technique for continuous systems and the Ping Pong algorithm is the way in which u(t) and v(t) are changed at the end of each iteration.

Inspection of equations (2.91) and (2.151) indicates that if $K_v = 1$, the change in v(t) in the Gradient Technique is identical with the change in v(t) in the Ping Pong algorithm. Thus if $K_v = 1$, the only real difference between the two techniques is the way u(t) is changed. Let us inspect this difference more closely.

From (2.21) and (2.22) it is clear that

$$H_u^m = \frac{\partial H(x^m, \lambda^m, u^m, v^m)}{\partial u} = - [< B, \lambda^m > + au^m] \tag{2.154}$$

Thus (2.150) can be written as

$$\delta_u^m(t) = \begin{cases} U_2 - u^m(t) & \text{if} \quad - H_u^m > U_2 \\ - H_u^m & \text{if} \quad U_1 \le - H_u^m \le U_2 \\ U_1 - u^m(t) & \text{if} \quad - H_u^m < U_1 \end{cases} \tag{2.155}$$

From this it is clear that in the Gradient Technique $u^{m+1}(t)$ is determined by changing u(t) in a "direction" related to $-H_u^m$. On the other hand, in the Ping Pong algorithm, $u^{m+1}(t)$ is determined by minimizing $H(x^m,\lambda^m,u^{m+1},v^{m+1})$ with respect to u^{m+1}. To carry the comparison of these two techniques further, it is convenient to define

$$\bar{\delta}_u^m(t) = \begin{cases} U_2 - u^m(t) & \text{if} \quad - < \lambda^m, B > > U_2 \\ - \frac{< \lambda^m, B >}{a} - u^m(t) & \text{if} \quad U_1 \le - \frac{< \lambda^m, B >}{a} \le U_2 \\ U_1 - u^m(t) & \text{if} \quad - < \lambda^m, B > < U_1 \end{cases} \tag{2.156}$$

From (2.93) it is seen that $\bar{\delta}_u^m(t)$ is the change in control between the m and m+1st iterations of the Ping Pong algorithm. In the Ping Pong algorithm

$$u^{m+1}(t) = K_u^m \bar{\delta}_u^m(t) + u^m(t) \tag{2.157}$$

where $K_u^m = 1$ while in the Gradient Technique

$$u^{m+1}(t) = K_u^m \delta_u^m(t) + u^m(t) \qquad (2.158)$$

where $0 \leq K_u^m \leq 1$.

Thus, there appear to be two basic differences between the Gradient Technique and the Ping Pong algorithm. The first of these is the difference between $\bar{\delta}_u^m(t)$ in (2.156) and $\delta_u^m(t)$ in (2.150). $\bar{\delta}_u^m(t)$ and $\delta_u^m(t)$ are associated with the "direction" of change or with the "shape" of change in u(t) between two successive iterations.

The more important difference between the two techniques is associated with the step size K_u^m which determines the magnitude of change in u(t). In the Gradient Technique, if K_u^m is too large, solutions will diverge rather than converge. It is reasonable to assume that the same thing can happen in the Ping Pong algorithm. That is, since K_u^m is fixed at 1 in the Ping Pong algorithm, changes in u(t) on successive iterations may be too large. It is believed that this is the reason why the algorithm has not worked satisfactorily.

An area for further study would be an investigation of the convergence properties of the Ping Pong algorithm with step sizes generated by the methods described in section 2.12.

2.14 Problems Investigated

This section contains descriptions and discussions of the Minimax Problems that we have studied. The first four problems studied were test problems designed to perform two functions:

1. To "test" the Minimax Principle. By picking relatively simple problems for which it is possible to guess certain properties of the solutions, we can see if the actual solutions have these same properties.

2. To act as preliminary test problems for new computational algorithms.

In the rest of the problems considered, the plant has been the rigid-body equations for Model Vehicle No. 2. The coefficients in these equations have been kept constant at the values they attain at max q (t = 80 seconds). The following equation has been used as a measure of the bending moment at the first staging point (90 meters):

$$\text{B.M.} = 2.244\ (\alpha_w - 4.05\phi - .00193\dot{z} + 5.05\beta) \times 10^6 \text{ Kg - M.} \qquad (2.159)$$

Unfortunately, it has been recently noted that this equation is incorrect. (see appendix B). The curves obtained using it are included anyway. It should be noted that the solutions to the problems considered (with the exception of 2.10) do not depend upon the bending moment expression and are therefore correct.

All variables from problem 2.5 on are in MKS units and radians.

Problem 2.1

Given:

$$\dot{x}_1 = -2x_1 + x_2 + v \qquad x_1(0) = 1$$

$$\dot{x}_2 = -x_2 + u \qquad x_2(0) = 1$$

$$J = \int_0^1 (x_1^2 + .5u^2)\, dt \qquad |v| \leq 1$$

From the Minimax Principle:

$$H = x_1^2 + .5u^2 + \lambda_1(-2x_1 + x_2 + v) + \lambda_2(-x_2 + u)$$

$$\dot{\lambda}_1 = 2\lambda_1 - 2x_1 \qquad \lambda_1(1) = 0$$

$$\dot{\lambda}_2 = -\lambda_1 + \lambda_2 \qquad \lambda_2(1) = 0$$

$$u^* = -\lambda_2$$

$$v^* = \text{sgn}\ \lambda_1$$

The solution to this problem obtained using the Ping Pong algorithm is shown in Figure 2.3. Since the natural frequencies of the plant are

both negative real numbers (-1 and -2) one would expect, for the initial conditions used, that the worst v(t) would be $v^*(t) \equiv 1$. Figure 2.3 shows that this is indeed true.

Problem 2.2

This is the same as problem 2.1 except that the initial conditions are picked as

$$x_1(0) = 0$$

$$x_2(0) = 0$$

A solution to this problem is given in Figure 2.4. From the symmetry of the problem it is clear that there is another solution with curves being the negative of those shown in Figure 2.4. The solution obtained from the digital computer, of course, will depend upon the initial guesses picked to start the algorithm (Ping Pong) used to solve the problem.

Problem 2.3

Given:

$$\dot{x}_1 = x_2 \qquad x_1(0) = 1$$

$$\dot{x}_2 = -49x_1 - x_2 + u + v \qquad x_2(0) = 1$$

$$J = \int_0^1 (x_1^2 + .5u^2)\, dt \qquad |v| \leq 1$$

From the Minimax Principle:

$$H = x_1^2 + .5u^2 + \lambda_1 x_2 + \lambda_2(-49x_1 - x_2 + u + v)$$

$$\dot{\lambda}_1 = -2x_1 + 49\lambda_2 \qquad \lambda_1(1) = 0$$

$$\dot{\lambda}_2 = -\lambda_1 + \lambda_2 \qquad \lambda_2(1) = 0$$

$$u^* = -\lambda_2$$

$$v^* = \text{sgn}\, \lambda_2$$

Figure 2.3

Solution Curves for Problem 2.1

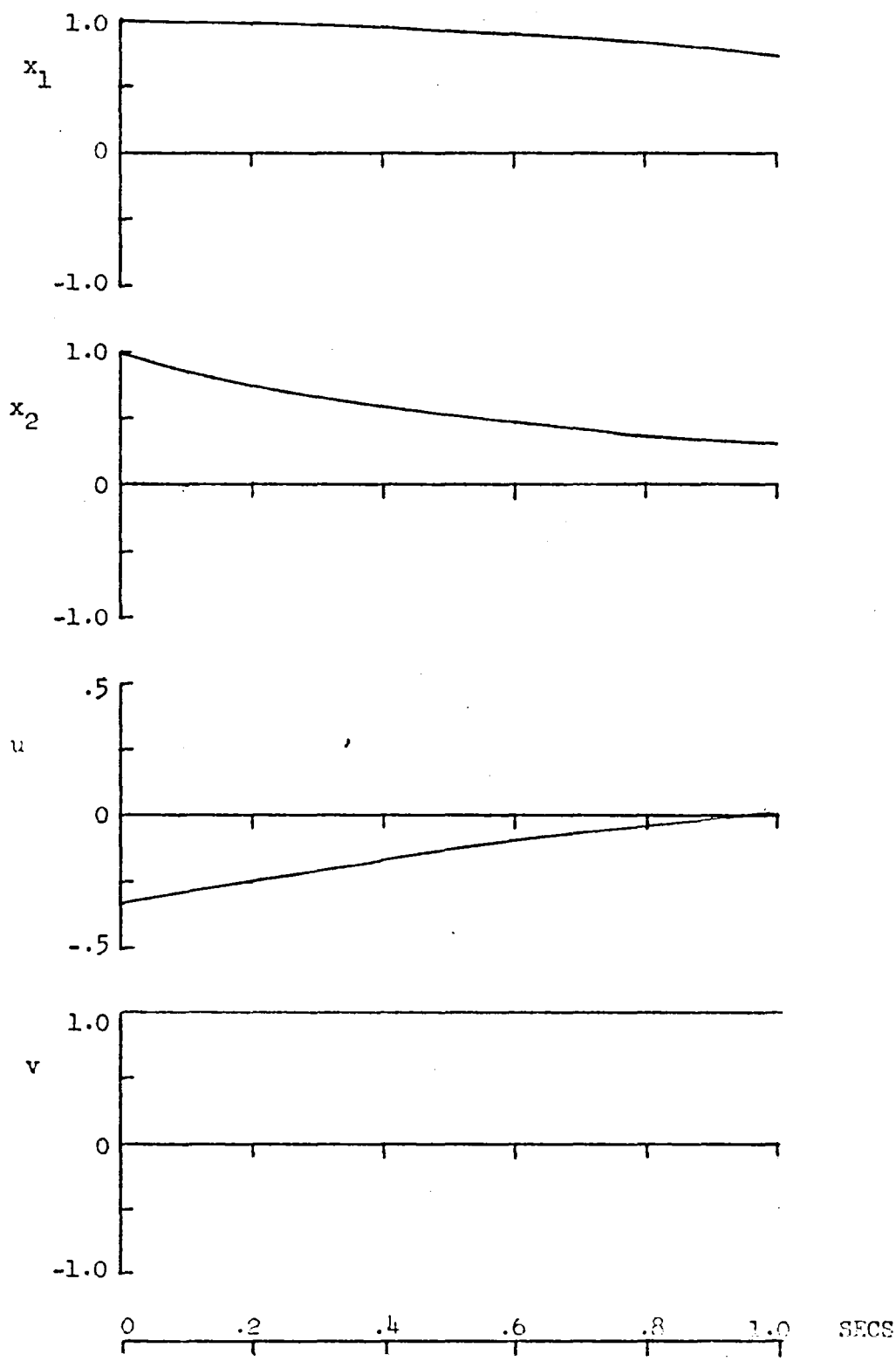
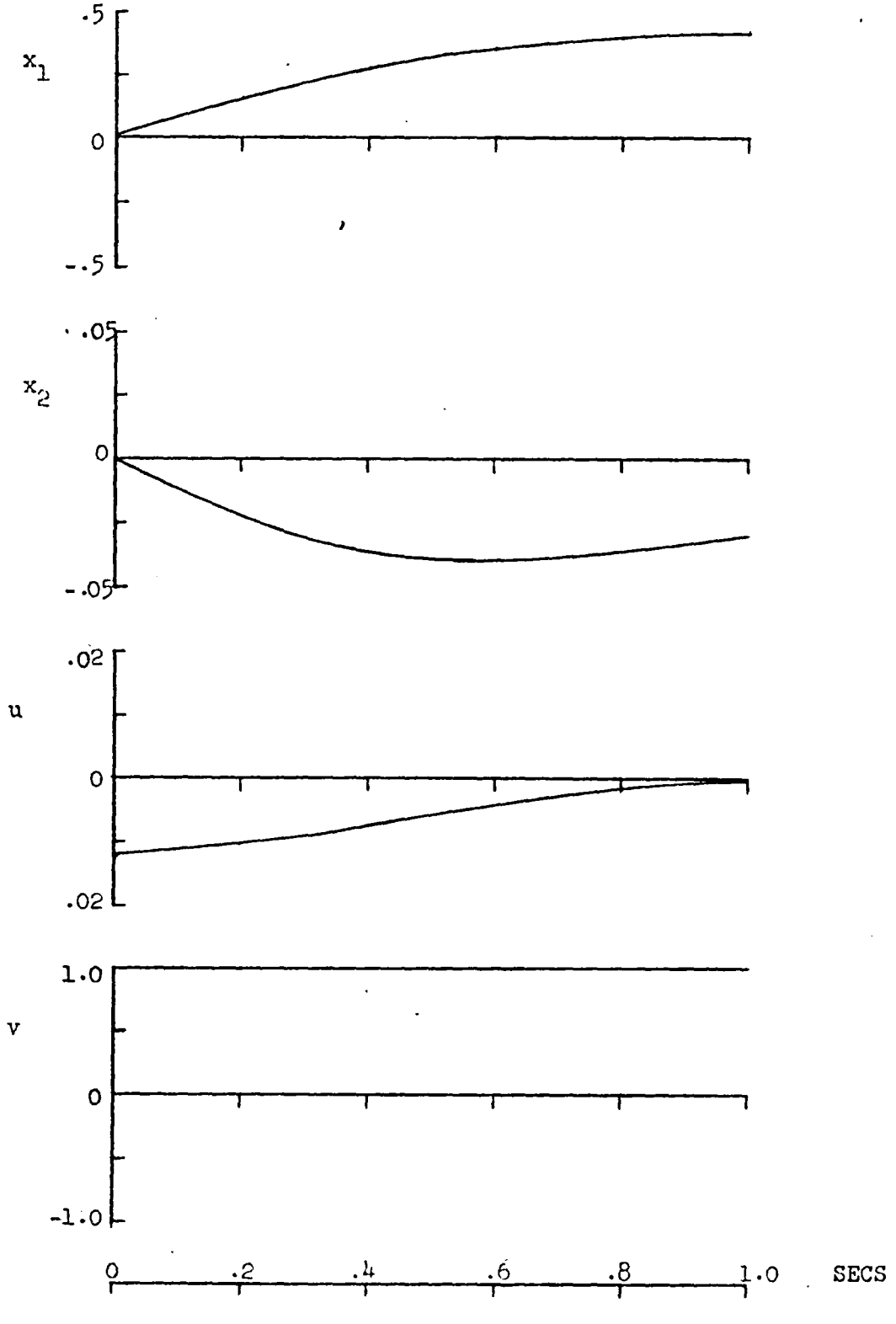


Figure 2.4

Solution Curves for Problem 2.2



Note that the plant is highly underdamped with a resonant frequency of approximately 1 cps. The solution to this problem is shown in Figure 2.5. The reason why x_2 becomes so large is because it is not included in the performance index. Note that u^* and v^* are always of opposite sign as one would anticipate from an inspection of the plant equations. This solution was first obtained using the Ping Pong algorithm.

Problem 2.4

Given

$$\dot{x}_1 = x_2 \qquad x_1(0) = 1$$

$$\dot{x}_2 = -49x_1 + x_2 + u + v \qquad x_2(0) = 1$$

$$J = \int_0^1 (x_1^2 + .5u^2)\, dt \qquad |v| \leq 1$$

From the Minimax Principle:

$$H = x_1^2 + .5u^2 + \lambda_1 x_2 + \lambda_2(-49x_1 + x_2 + u + v)$$

$$\dot{\lambda}_1 = -2x_1 + 49\lambda_2 \qquad \lambda_1(1) = 0$$

$$\dot{\lambda}_2 = -\lambda_1 - \lambda_2 \qquad \lambda_2(1) = 0$$

The only difference between this problem and problem 2.3 is the change in the sign of the coefficient of x_2 in the $\dot{x}_2$ equation. Hence the poles associated with this plant are in the right-half plane making the plant unstable. The solution to this problem can be found in Figure 2.6. This problem was also first solved with the Ping Pong algorithm.

Problem 2.5

Given:

$$x_1 = \phi,\ x_2 = \dot{\phi},\ x_3 = \dot{z}$$

$$x_1(0) = x_2(0) = x_3(0) = 0$$

$$\dot{x}_1 = .0791x_1 - .000153x_3 - .448\beta + .0791\alpha_w$$

Figure 2.5

Solution Curves for Problem 2.3

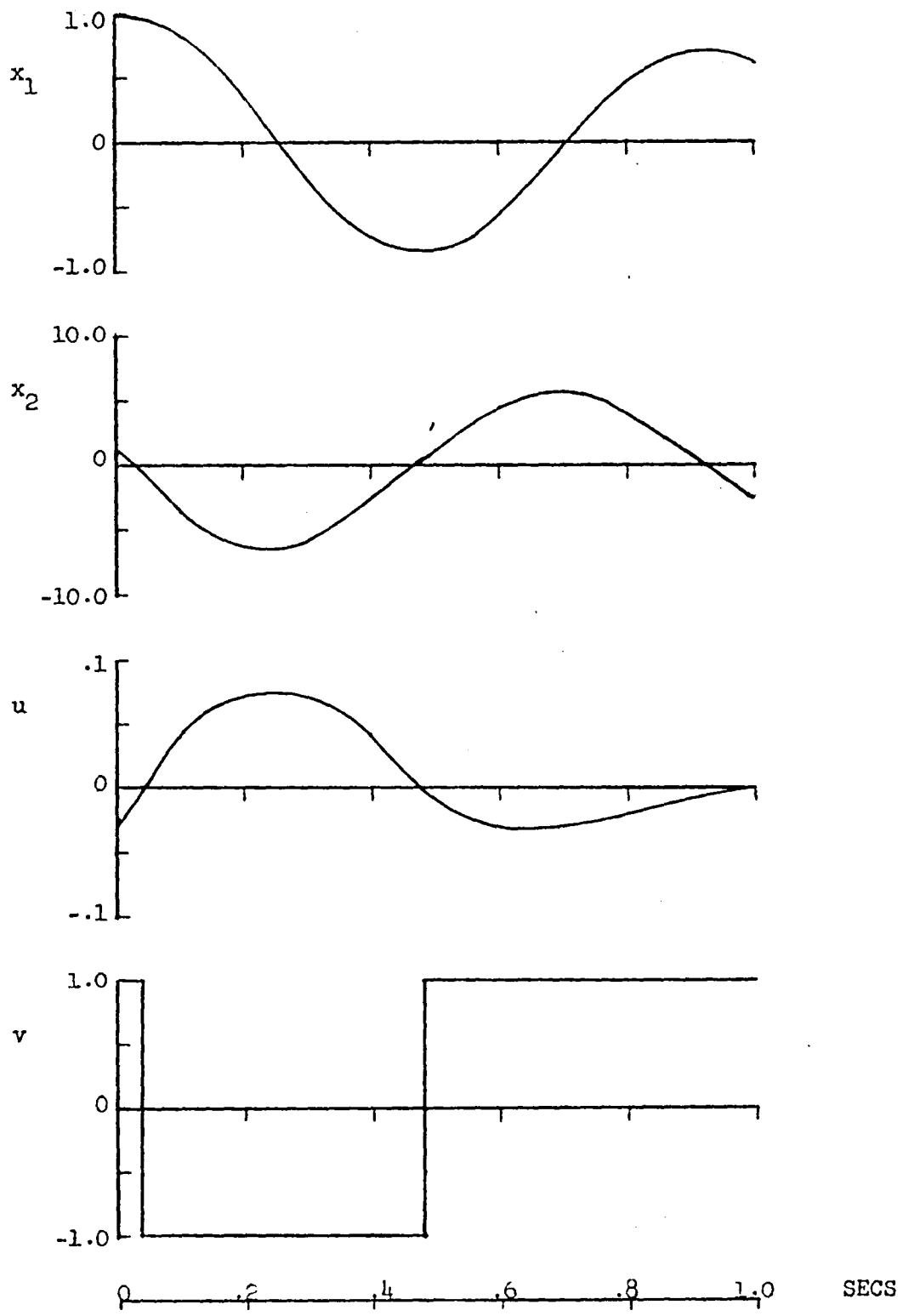
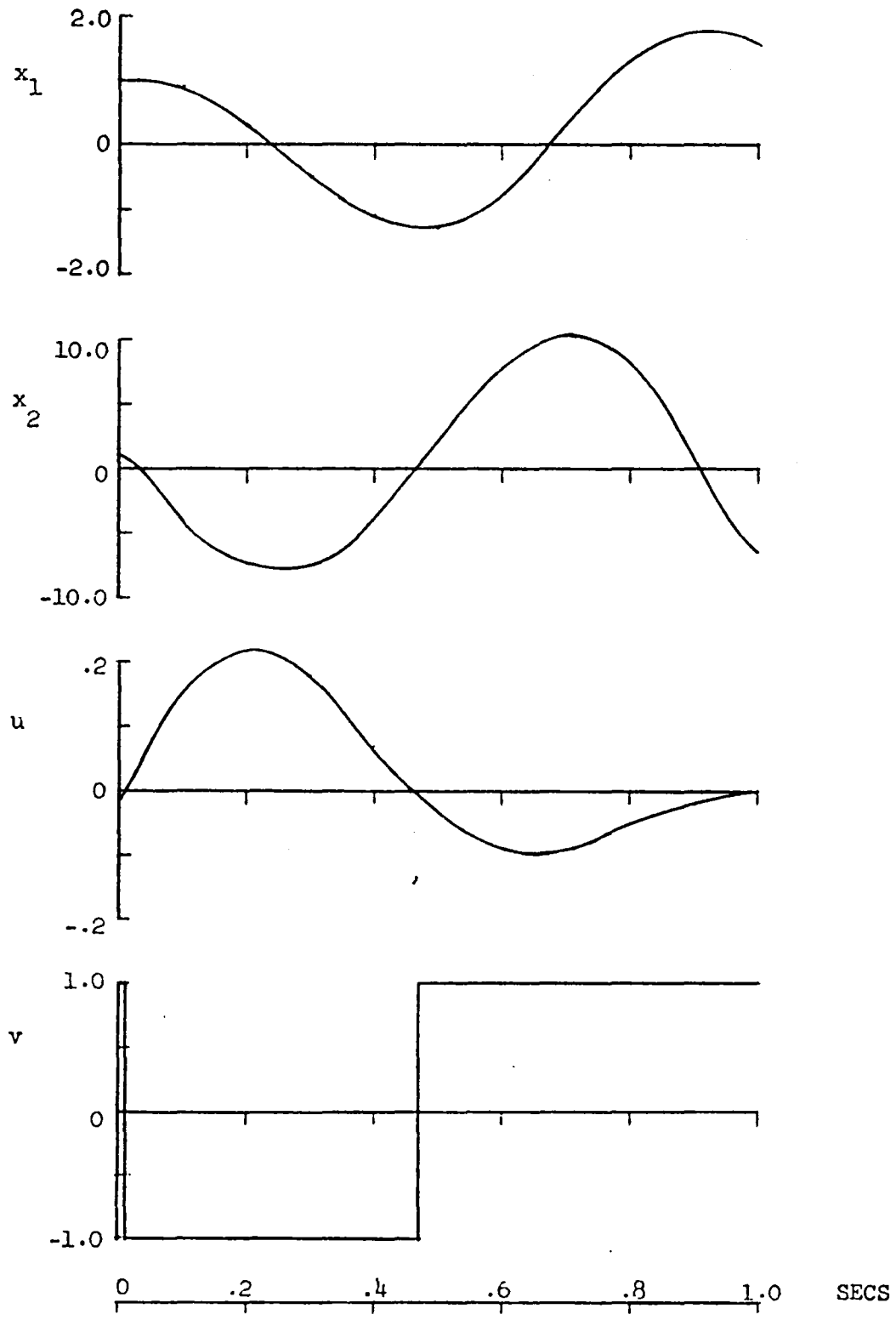


Figure 2.6

Solution Curves for Problem 2.4



$$\dot{x}_3 = 26.59x_1 - .0108x_3 + 10.93\beta + 5.59\alpha_w$$

$$0 \le \alpha_w \le .261 \qquad (15^\circ)$$

$$|\beta| \le .087 \qquad (5^\circ)$$

$$J = \frac{1}{2}\int_0^T (x_1^2 + x_2^2 + x_3^2 + \beta^2)\, dt$$

From the Minimax Principle:

$$H = \frac{1}{2}(x_1^2 + x_2^2 + x_3^2 + \beta^2) + \lambda_1 x_2 + \lambda_2(.0791x_1 - .000153x_3 - 448\beta + .0791\alpha_w) + \lambda_3(26.59x_1 - .0108x_3 + 10.93\beta + 5.59\alpha_w)$$

$$\dot{\lambda}_1 = x_1 - .0791\lambda_2 - 26.59\lambda_3 \qquad \lambda_1(T) = 0$$

$$\dot{\lambda}_2 = -x_2 - \lambda_1 \qquad \lambda_2(T) = 0$$

$$\dot{\lambda}_3 = -x_3 + .000153\lambda_2 + .0108\lambda_3 \qquad \lambda_3(T) = 0$$

$$\beta^* = .087 \text{ sat } \left(\frac{.448\lambda_2 - 10.93\lambda_3}{.087}\right)$$

$$\alpha_w^* = .1305\,(1 + \text{sgn}\,(.0791\lambda_2 + 5.59\lambda_3))$$

This was the first problem investigated using the equations of motion of Model Vehicle No. 2. The performance index is chosen in an attempt to meet goal 1 (section 2.5) which is to keep the vehicle reasonably close to the nominal trajectory all the time. β^2 is included in the performance index so that $\beta(t)$ will not be bang-bang.

Numerous attempts were made to solve this problem with the Ping Pong algorithm for various values of T. For T greater than 1 or 2 seconds, convergent solutions could not be obtained. In addition to the step size problem discussed at the end of section 2.13, several other factors are thought to have contributed to the convergence difficulties. One factor may be the specific performance index used. Since x_3 (drift rate) would, in general, be much larger than x_1 and x_2, the equal weightings given to these three state variables in the performance index is unreasonable.

That is, the performance index for this problem is approximately

$$\frac{1}{2}\int_0^T x_3^2 \, dt$$

A more reasonable performance index might be picked by weighting each term in the performance index by the reciprocal of its expected peak value. For example, if the peak values of $|x_1|$, $|x_2|$, $|x_3|$ and $|\beta|$ are roughly $|x_{1\,max}|$, $|x_{2\,max}|$, $|x_{3\,max}|$ and $|\beta_{max}|$ one might use a performance index of the form

$$J = \frac{1}{2}\int_0^T \left[\frac{x_1^2}{(x_{1\,max})^2} + \frac{x_2^2}{(x_{2\,max})^2} + \frac{x_3^2}{(x_{3\,max})^2} + \frac{\beta^2}{(\beta_{max})^2} \right] dt$$

This approach is used in choosing the performance indices of all following problems.

Another factor thought to affect convergence was the upper bound on α_w. Lowering this bound from .261 radians to .120 radians had little effect on convergence with the Ping Pong algorithm. That is, convergent solutions still could not be obtained. A solution to problem 5 for T = 1 second is shown in Figures 2.7a and 2.7b. The solution was first obtained using the Ping Pong algorithm.

Problem 2.6

Given:

$$x_1 = \phi, \; x_2 = \dot{\phi}, \; x_3 = \dot{z}$$

$$x_1(0) = x_2(0) = x_3(0) = 0$$

$$\dot{x}_2 = .0791x_1 - .000153x_3 - .448\beta + .0791\alpha_w$$

$$\dot{x}_3 = 26.59x_1 - .0108x_3 + 10.93\beta + 5.59\alpha_w$$

$$0 \le \alpha_w \le .120$$

$$|\beta| \le .087$$

Figure 2.7a

Solution Curves for Problem 2.5

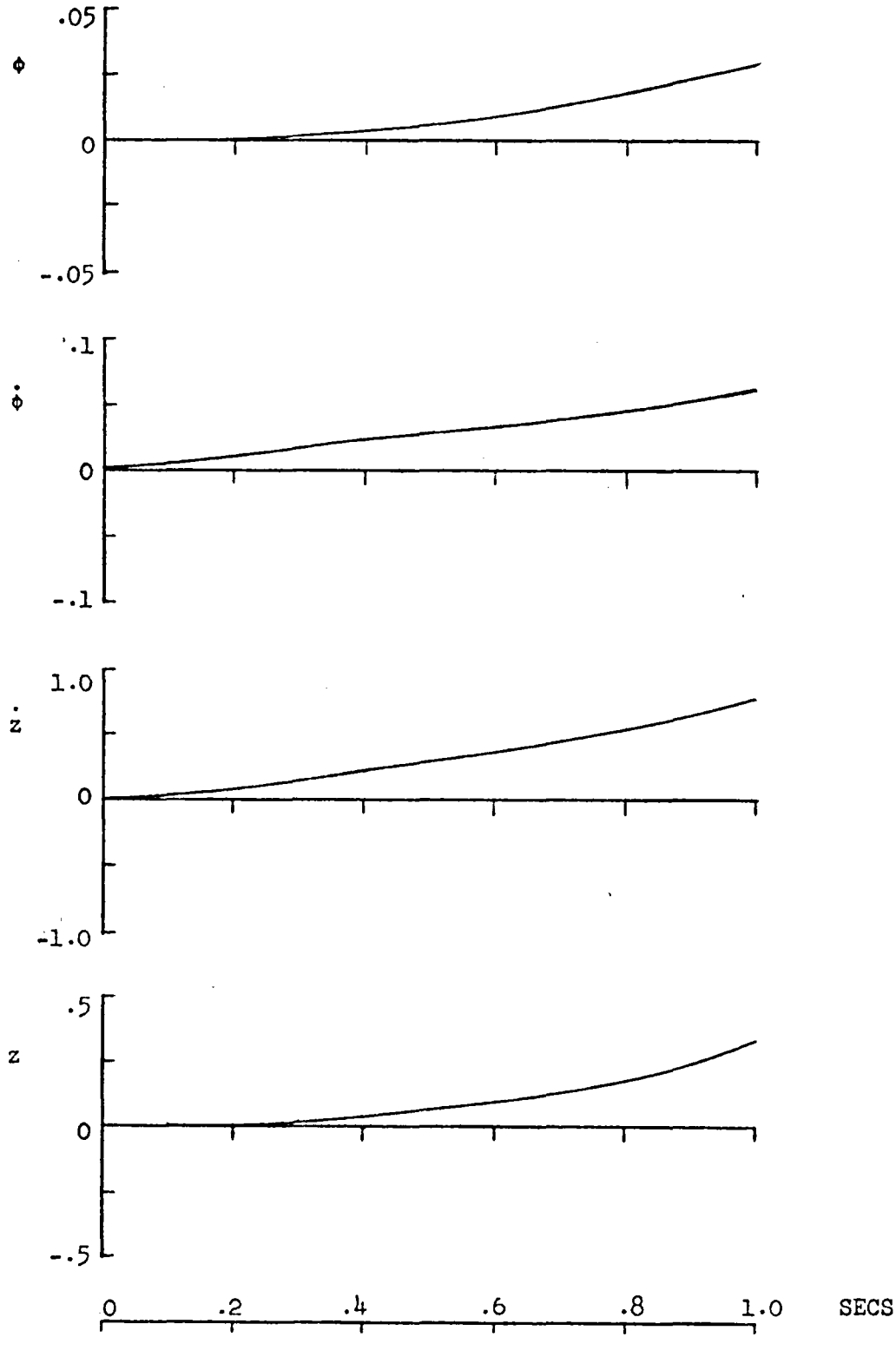
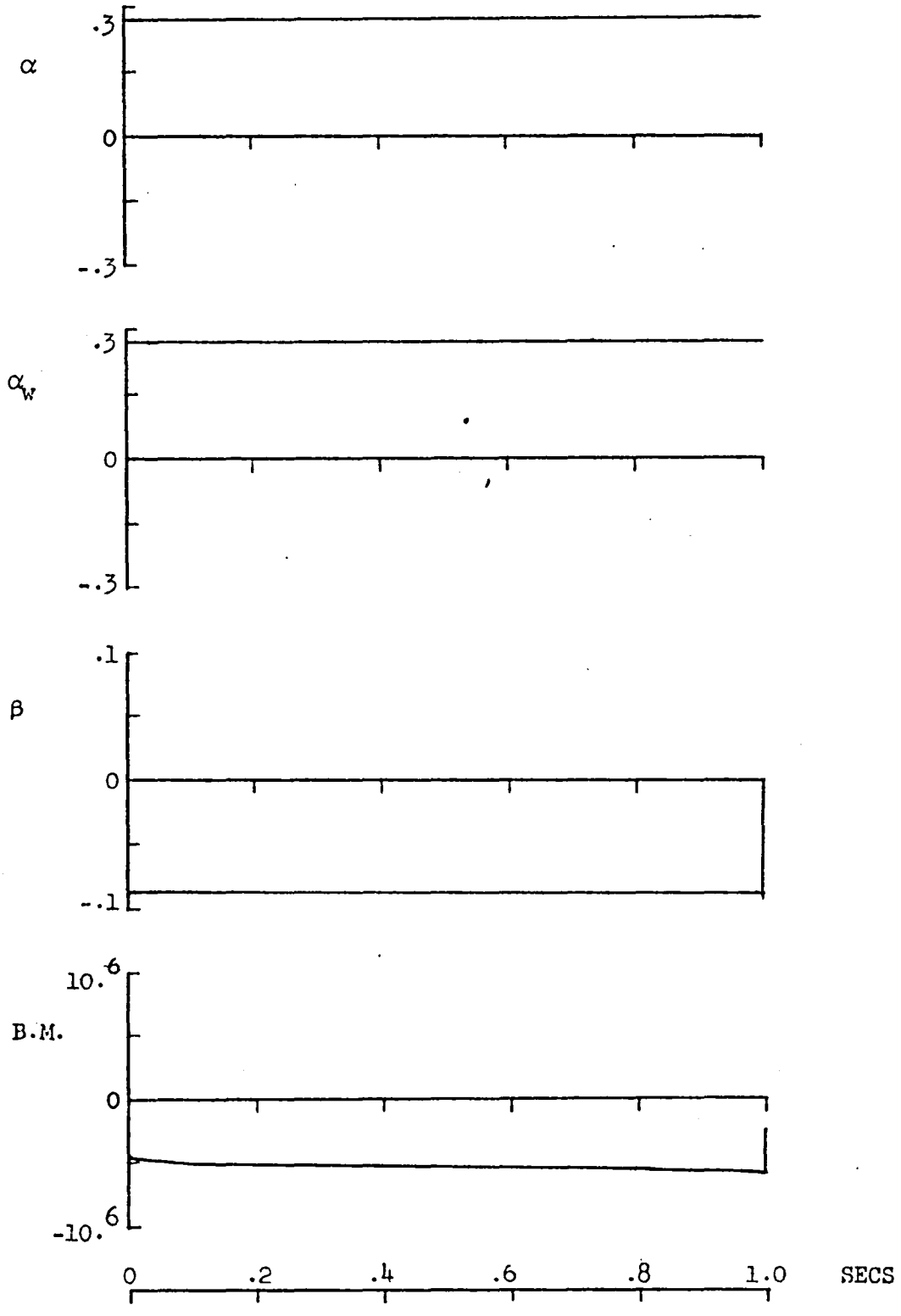


Figure 2.7b

Solution Curves for Problem 2.5



$$J = \frac{1}{2}\int_0^T (x_1^2 + x_2^2 + 10\beta^2 + .004x_3)\, dt \qquad T = 5 \text{ seconds}$$

Note that

$$Z(T) = \int_0^T x_3\, dt + Z(0)$$

From the Minimax Principle:

$$H = \frac{1}{2}(x_1^2 + x_2^2 + 10\beta^2 + .004x_3) + \lambda_1 x_3 + \lambda_2(-.0791x_1 - .000153x_3 - .448\beta + .0791\alpha_w + \lambda_3(26.59x_1 - .0108x_3 + 10.93\beta + 5.59\alpha_w$$

$$\dot{\lambda}_1 = -x_1 - .0791\lambda_2 - 26.59\lambda_3 \qquad \lambda_1(T) = 0$$

$$\dot{\lambda}_2 = -x_2 - \lambda_1 \qquad \lambda_2(T) = 0$$

$$\dot{\lambda}_3 = .000153\lambda_2 + .0108\lambda_3 - .002 \qquad \lambda_3(T) = 0$$

$$\beta^* = .087 \text{ sat}\left\{\frac{.448\lambda_2 - 10.93\lambda_3}{.87}\right\}$$

$$\alpha_w^* = .06(1 + \text{sgn}\,(.0791\lambda_2 + 5.59\lambda_3))$$

The major difference between this problem and problem 2.5 is the new performance index. Since the lower bound on α_w is zero, it was expected that Z(T), the normal distance between the center of gravity of the vehicle and the reference trajectory, would be greater than zero. Hence it seemed reasonable to use Z(T) rather than $Z^2(T)$ in the performance index. If $Z^2(T)$ were to be used, an additional differential equation would have to be employed, raising the order of the system to four. This performance index was picked to try to keep $\phi(t)$ and $\dot{\phi}(t)$ small during flight (goal 1) and to make Z(T) small (goal 2). Again $\beta^2(t)$ is in the performance index to keep $\dot{\beta}(t)$ bounded (i.e., to prevent $\beta(t)$ from being bang-bang).

It was assumed that

$$|x_{1\,\max}| \approx .1$$

$$|x_{2\ max}| \approx .1$$
$$|x_{3\ max}| \approx 2.5$$
$$|\beta_{max}| \approx .1$$

Using the approach indicated in problem 2.5, these numbers give

$$\int_0^T \left(\frac{x_1^2}{.01} + \frac{x_2^2}{.01} + \frac{\beta^2}{.01} + \frac{x_3}{2.5}\right) dt = 100 \int_0^T (x_1^2 + x_2^2 + \beta^2 + .004x_3)\, dt$$

A constant times a performance index has no effect on trajectories so $\frac{1}{2}$ was used instead of 100. β^2 received an additional weighting of 10 to keep $\dot{\beta}$ small.

The Ping Pong algorithm was applied to this problem first. No convergent solution could be obtained with it. In fact, the Ping Pong algorithm did not work for any problems following 2.5.

The solution to this problem, obtained via Quasilinearization, is shown in Figures 2.8a and 2.8b. Note that Z(T) is positive as expected. Note also that $\beta(t)$ never saturates.

Problem 2.7

This problem is the same as problem 2.6, except for the following changes:

1. T = 10 seconds
2. The coefficient of $\beta(t)$ in the performance index is reduced from 10 to 1 since $\beta(t)$ did not saturate in the last problem.

The solution to this problem, obtained with Quasilinearization, is shown in Figures 2.9a and 2.9b. This problem shows that Z(T) can indeed become negative. Hence the performance index must be changed.

Problem 2.8

This problem is the same as problem 2.7, except that the magnitude of $\beta(t)$ is not constrained. The solution to this problem is shown in Figures 2.10a and 2.10b. Note that Z(T) is more negative in this problem

Figure 2.8a

Solution Curves for Problem 2.6

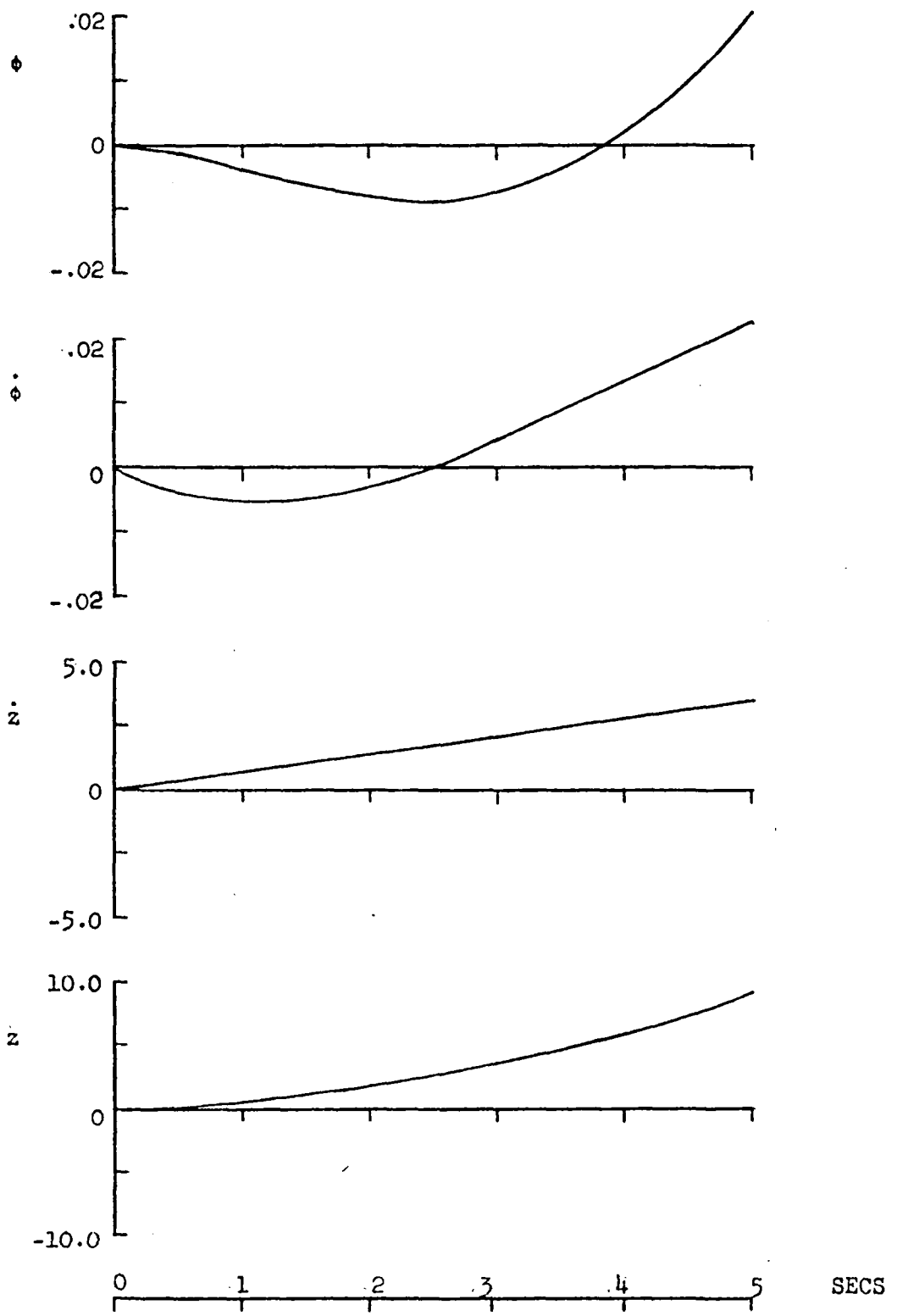


Figure 2.8b

Solution Curves for Problem 2.6

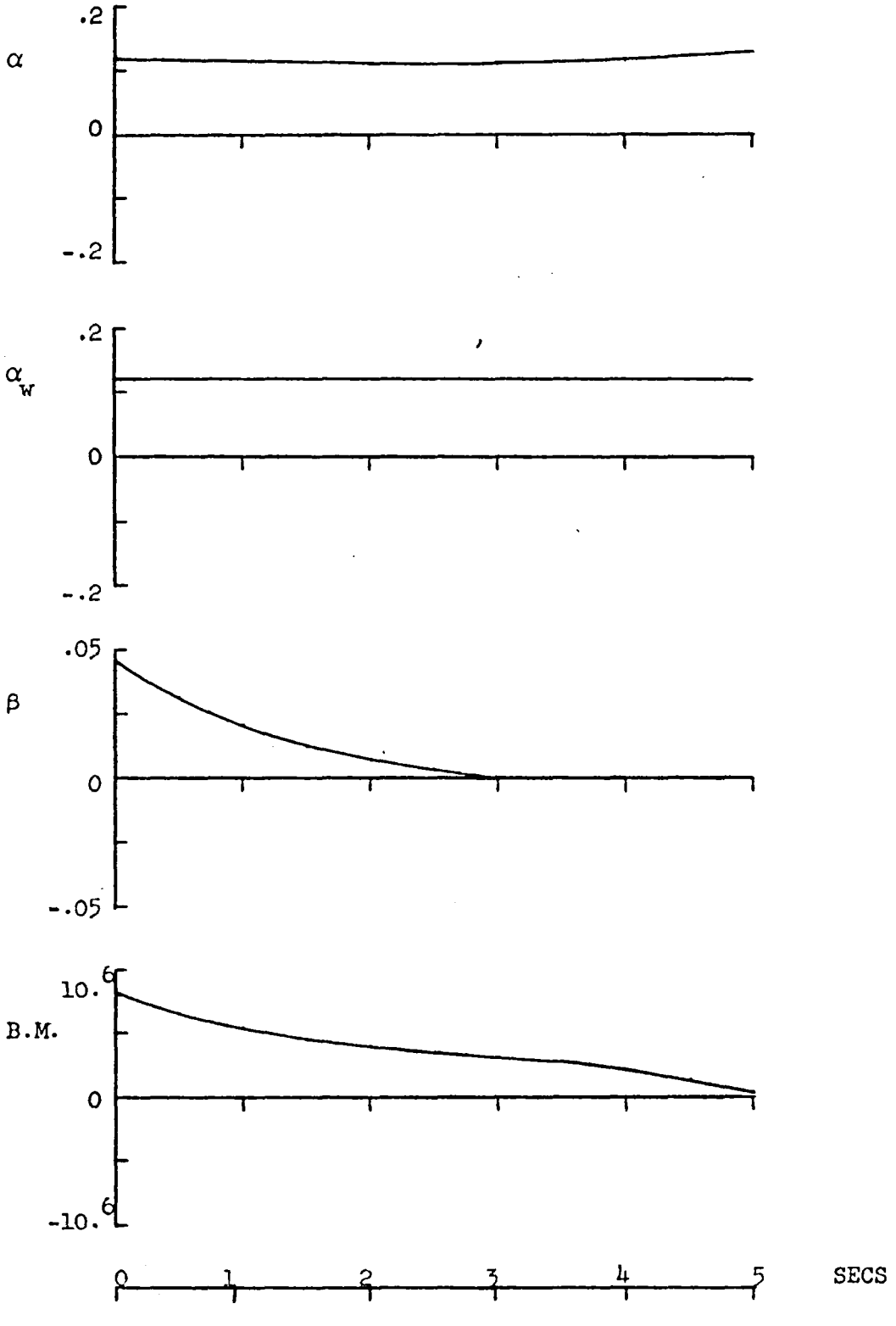


Figure 2.9a

Solution Curves for Problem 2.7

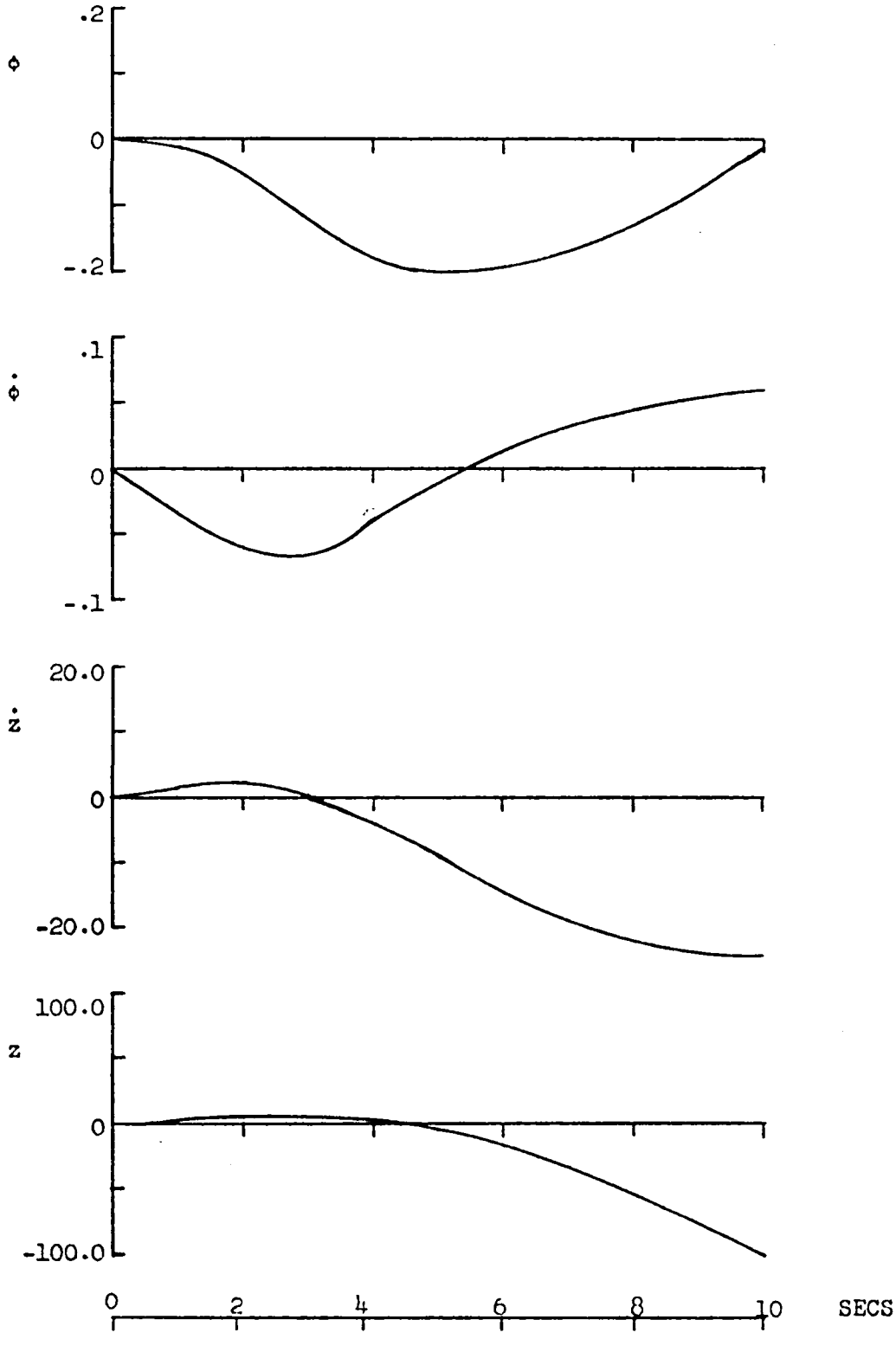


Figure 2.9b

Solution Curves for Problem 2.7

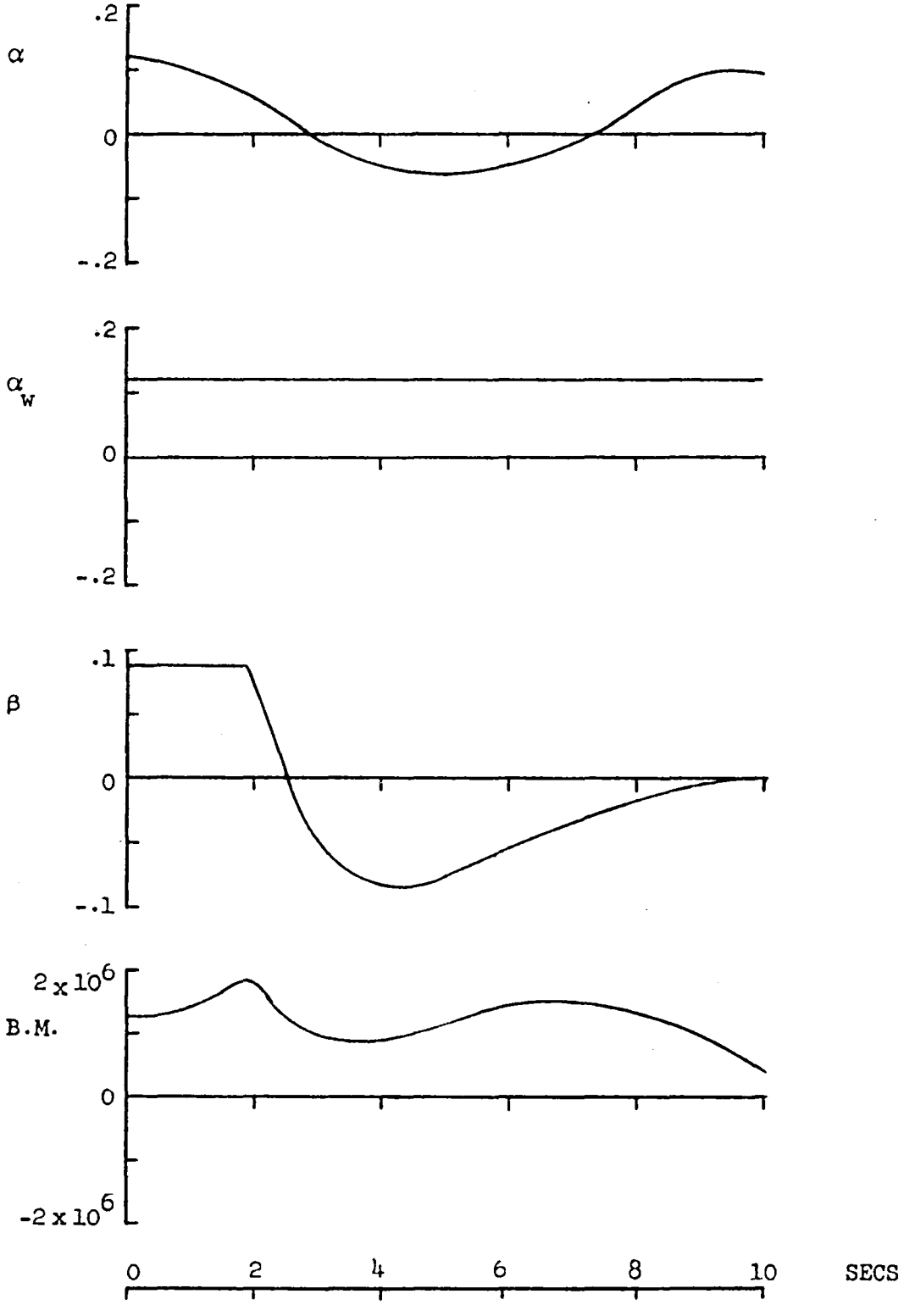


Figure 2.10a

Solution Curves for Problem 2.8

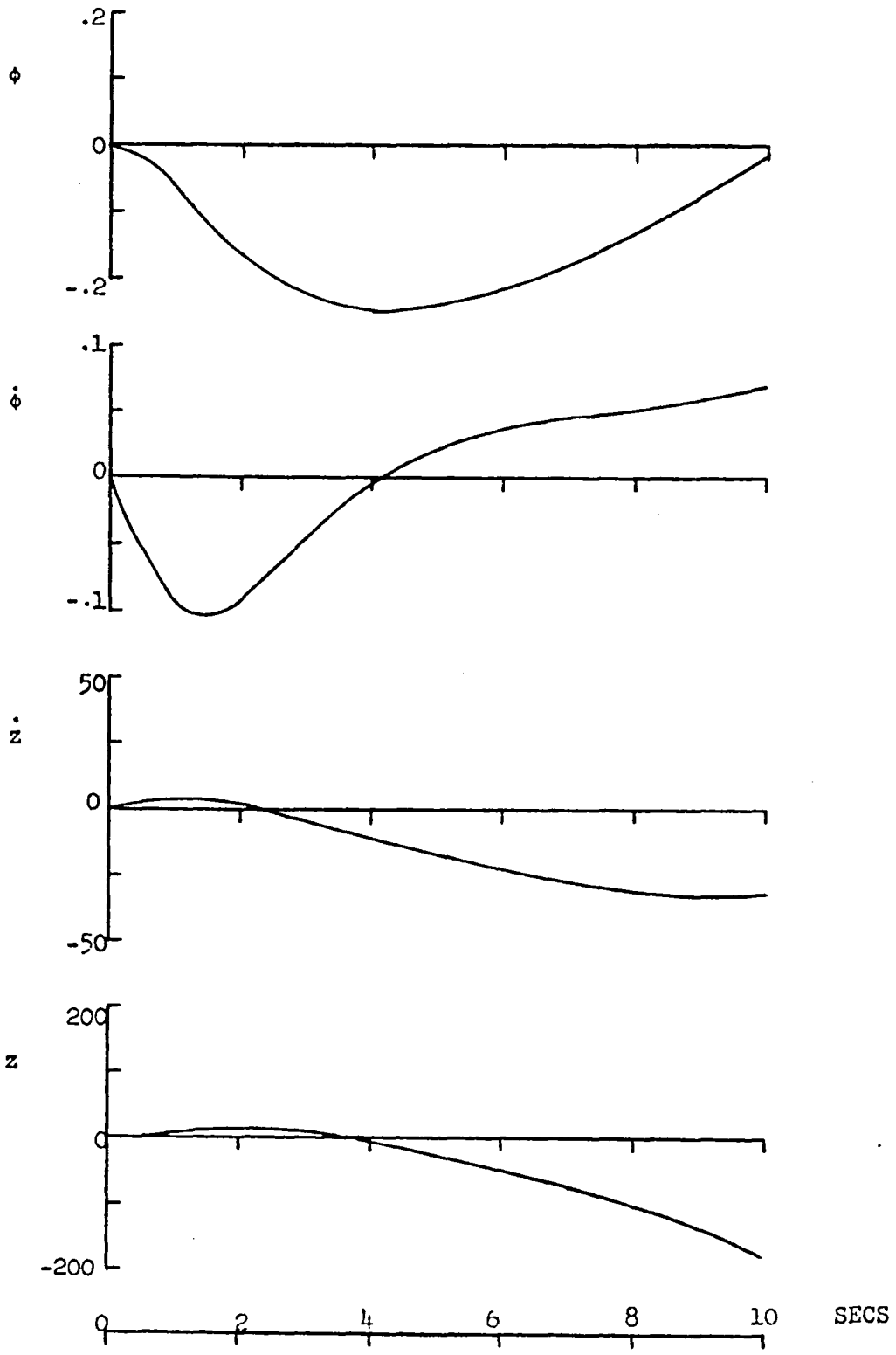
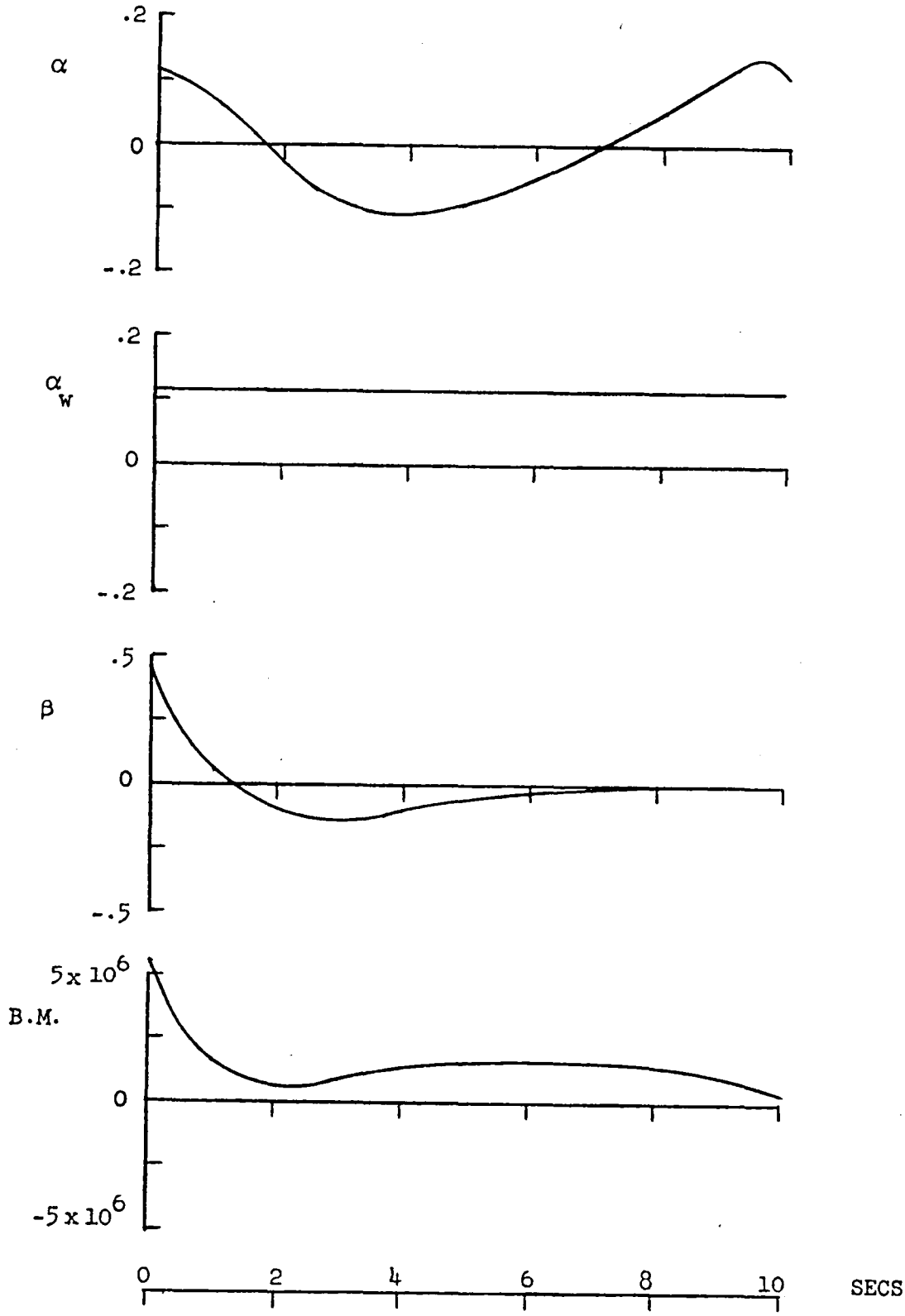


Figure 2.10b

Solution Curves for Problem 2.8



than it was in problem 2.7. This, of course, is because $\beta(t)$ is not constrained.

Problem 2.9

Given:

$$x_1 = \phi,\ x_2 = \dot{\phi},\ x_3 = \dot{z}$$

$$x_1(0) = x_2(0) = x_3(0) = 0$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = .0791x_1 - .000153x_3 - .448\beta + .07\alpha_w$$

$$\dot{x}_3 = 26.59x_1 - .0108x_3 + 10.93\beta + 5.59\alpha_w$$

$$\alpha_w = .087$$

$$J = \frac{1}{2}\int_0^{10} (x_1^2 + x_2^2 + .0004x_3^4 + \beta^2)\, dt$$

From the Maximum Principle:

$$H = \frac{1}{2}(x_1^2 + x_2^2 + .0004x_3^4 + \beta^2) + \lambda_1 x_2 + \lambda_2(.0791x_1 - .000153x_3 - .448\beta + .07\alpha_w) + \lambda_3(26.59x_1 - .0108x_3 + 10.93\beta + 5.59\alpha_w)$$

$$\dot{\lambda}_1 = -x_1 - .0791\lambda_2 - 26.59\lambda_3 \qquad \lambda_1(10) = 0$$

$$\dot{\lambda}_2 = -x_2 - \lambda_1 \qquad \lambda_2(10) = 0$$

$$\dot{\lambda}_3 = -.0008x_3^3 + .000153\lambda_2 + .0108\lambda_3 \qquad \lambda_3(10) = 0$$

$$\beta^* = .448\lambda_2 - 10.93\lambda_3$$

Note the following:

1. Since α_w is given, this is not a Minimax Problem but a minimum problem.
2. There are no bounds on $\beta(t)$.
3. The coefficient of α_w in the $\dot{x}_2$ differential equation is .070 while for Model Vehicle No. 2 it should be .079.

4. There is a term of higher than second power in the performance index, giving rise to a nonlinear term in the adjoint equations.
5. This problem was written as a homework exercise for the students in one of the advanced automatic control courses. The purpose of this problem was to serve as an example on which Quasilinearization was to be applied. It is felt that the problem will be of interest even though it is not a Minimax Problem.

M.S.F.C. is no doubt aware of the fact that to obtain solutions to the problems discussed in this chapter is by no means a simple matter. Each solution that has been presented thus far has been obtained only after considerable time and effort have been spent on it. This particular problem provides a graphic example.

Six graduate students, including one researcher working on this contract, attacked this problem on an individual basis. It took more than six weeks for any one of them to obtain a solution.

The reason for the computational difficulties is now known. The set of equations composed of the $\dot{x}$ and the $\dot{\lambda}$ equations forms, in any linear plant, quadratic performance index problem, an unstable system. By including the nonlinear term x_3^3 in the $\dot{\lambda}_3$ equation, solutions to this set of equations become highly sensitive to initial conditions. Since initial conditions must be guessed to start Quasilinearization, a bad guess could cause the solution to the differential equations to blow up very rapidly. This in fact was the case. When solutions started to blow up, digital computer integration routines failed. Hence it became extremely difficult to obtain an accurate first solution to the differential equations. Without such a solution, Quasilinearization could not be employed.

The problem was finally solved as follows. A variable coefficient was placed in front of all nonlinear terms in the differential equations. At first this coefficient was made very small and a solution to the boundary value problem was obtained using Quasilinearization. The coefficient was then increased by a small amount. The new set of differential equations were again solved using Quasilinearization with the solution for the previous setting of the coefficient as the initial guess. The process was continued until the differential equations solved were the ones specified in the problem.

Solution curves for this problem are shown in Figure 2.11.

Problem 2.10

Given:

$$x_1 = \phi,\ x_2 = \dot{\phi},\ x_3 = \dot{z},\ x_4 = \beta$$

$$\dot{x}_1 = x_2 \qquad x_1(0) = 0$$

$$\dot{x}_2 = .0791x_1 - .000153x_3 - .448x_4 + .0791\alpha_w \qquad x_2(0) = 0$$

$$\dot{x}_3 = 26.59x_1 - .0108x_3 + 10.93x_4 + 5.59\alpha_w \qquad x_3(0) = 0$$

$$\dot{x}_4 = 1/\tau(\beta_c - x_4)$$

$$\text{B.M.} = (2.244\text{x}10^6)G \qquad |\beta_c| \le .087$$

$$0 \le \alpha_w \le .120$$

where

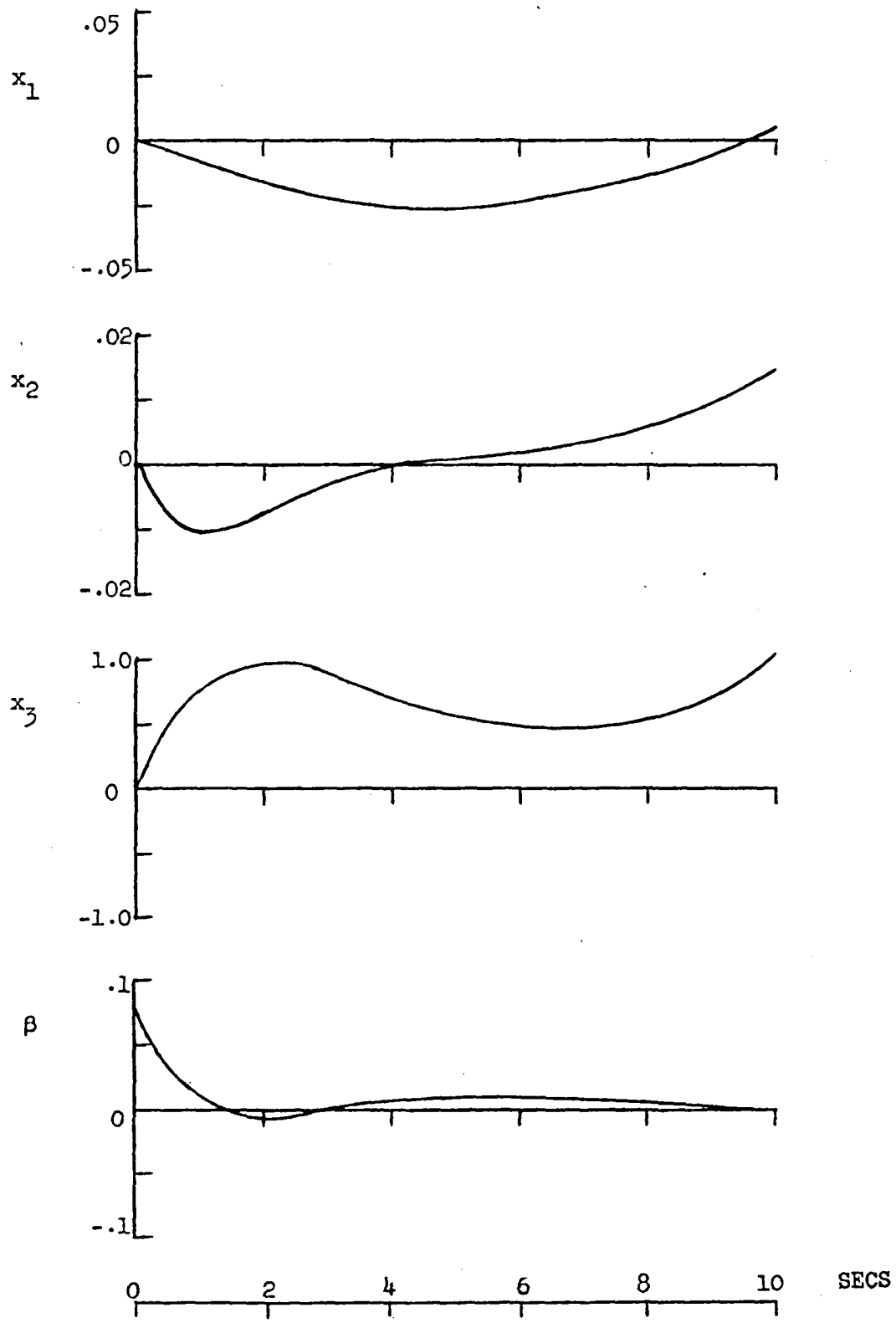
$$G = (5.05x_4 - 4.05x_1 + \alpha_w - .00193x_3)$$

$$J = \frac{1}{T}\int_0^T G^2\,dt + \frac{x_2^2(T)}{(.1)^2} + \frac{x_3^2(T)}{(5)^2}$$

The problem is to minimize J with respect to β_c and to maximize it with respect to α_w. Note the following:

Figure 2.11

Solution Curves for Problem 2.9



1. In order to apply the Minimax Principle as stated in Chapter III, both the differential equations and the performance index must be separable in control and disturbance. Since G is a function of β and α_w, J is not separable in β and α_w. Hence instead of minimizing with respect to β, minimization is done with respect to β_c. Note that

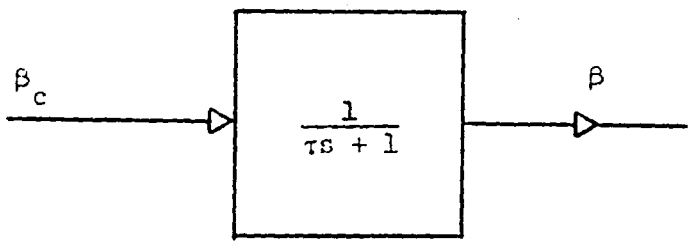


Figure 2.12

Block Diagram Relating β and β_c

might be thought of as a first approximation to engine gimbal dynamics

2. In this problem the performance index is the integral-squared value of the bending moment (goal 3) plus the final values of drift and attitude rates (goal 2).

Unfortunately, we have not been able to obtain a solution to this problem. Both Quasilinearization and the Gradient Techniques have been tried with no success. It is thought that there may be no solution to this particular problem.

Problem 2.11

This problem is the same as problem 2.6 except for the upper bound on α_w. For this problem

$$0 \leq \alpha_w \leq .144 \qquad (8.25 \text{ degrees})$$

The Minimax solution to this problem is shown in Figures 2.13a and 2.13b. The solution was obtained via the first Gradient Technique discussed in section 2.12.

Using the same β, the problem was rerun with an α_w provided by M.S.F.C. The results are shown in Figure 2.13c and 2.13d. α_w was chosen in the time interval 75 to 80 seconds on the curve provided by M.S.F.C. The largest value of this α_w is 8.25 degrees.

The value of the performance index with the M.S.F.C. α_w is .01714. Using the worst α_w (calculated from the Minimax Principle) the value of the Performance index is .03218. Hence, the worst α_w produces a higher valued performance index, as expected.

Problem 2.12

Given:

$$x_1 = \phi, \; x_2 = \dot{\phi}, \; x_3 = \dot{z}$$

$$x_1(0) = x_2(0) = x_3(0) = 0$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = .0791x_1 - .000153x_3 - .448\beta + .0791\alpha_w$$

$$\dot{x}_3 = 26.59x_1 - .0108x_3 + 10.93\beta + 5.59\alpha_w$$

$$|\beta| \leq .087$$

$$0 \leq \alpha_w \leq .144$$

$$x_2(T) = 0 \quad , \quad T = 5 \text{ seconds}$$

Figure 2.13a

Solution Curves for Problem 2.11
Using Worst α_w

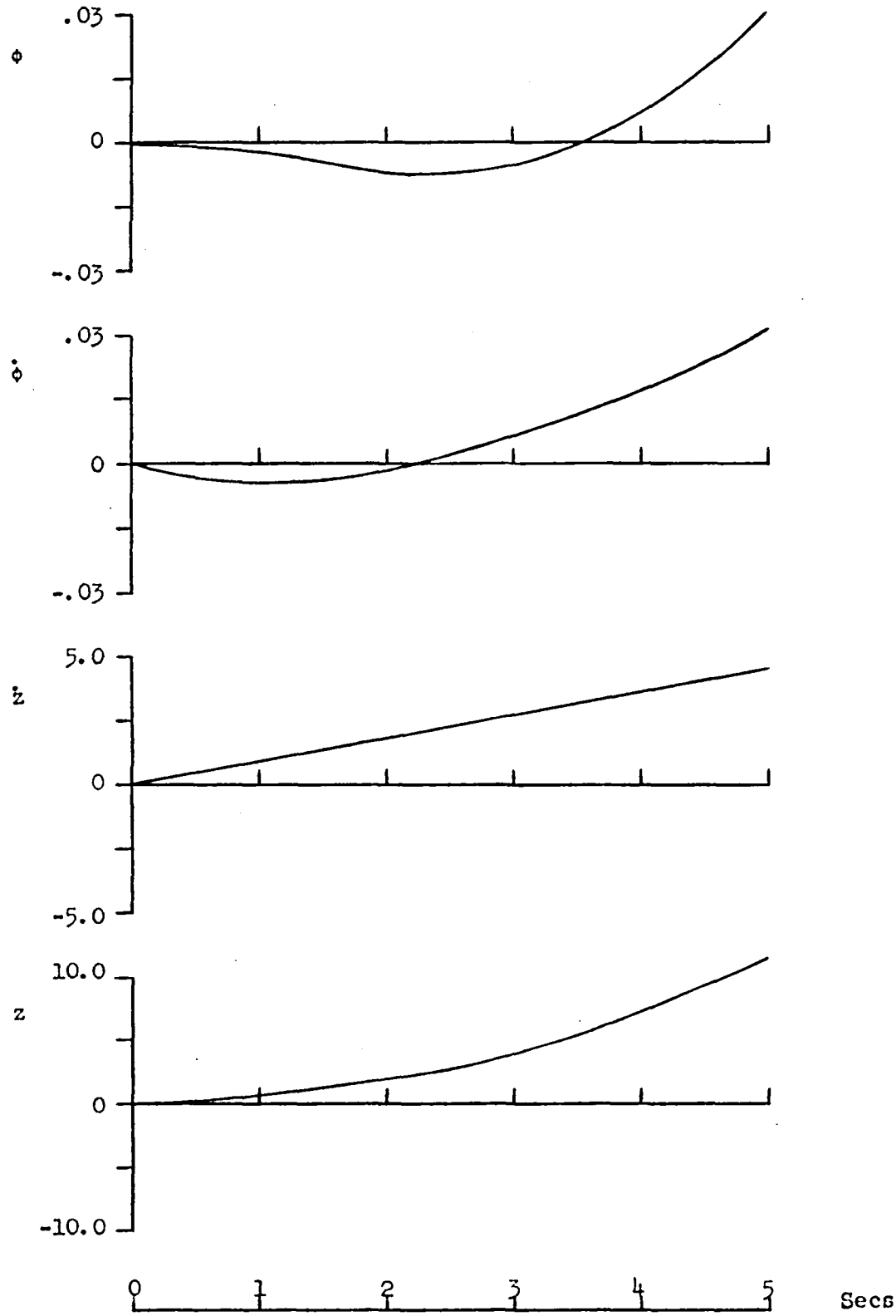


Figure 2.13b

Solution Curves for Problem 2.11
Using Worst α_w

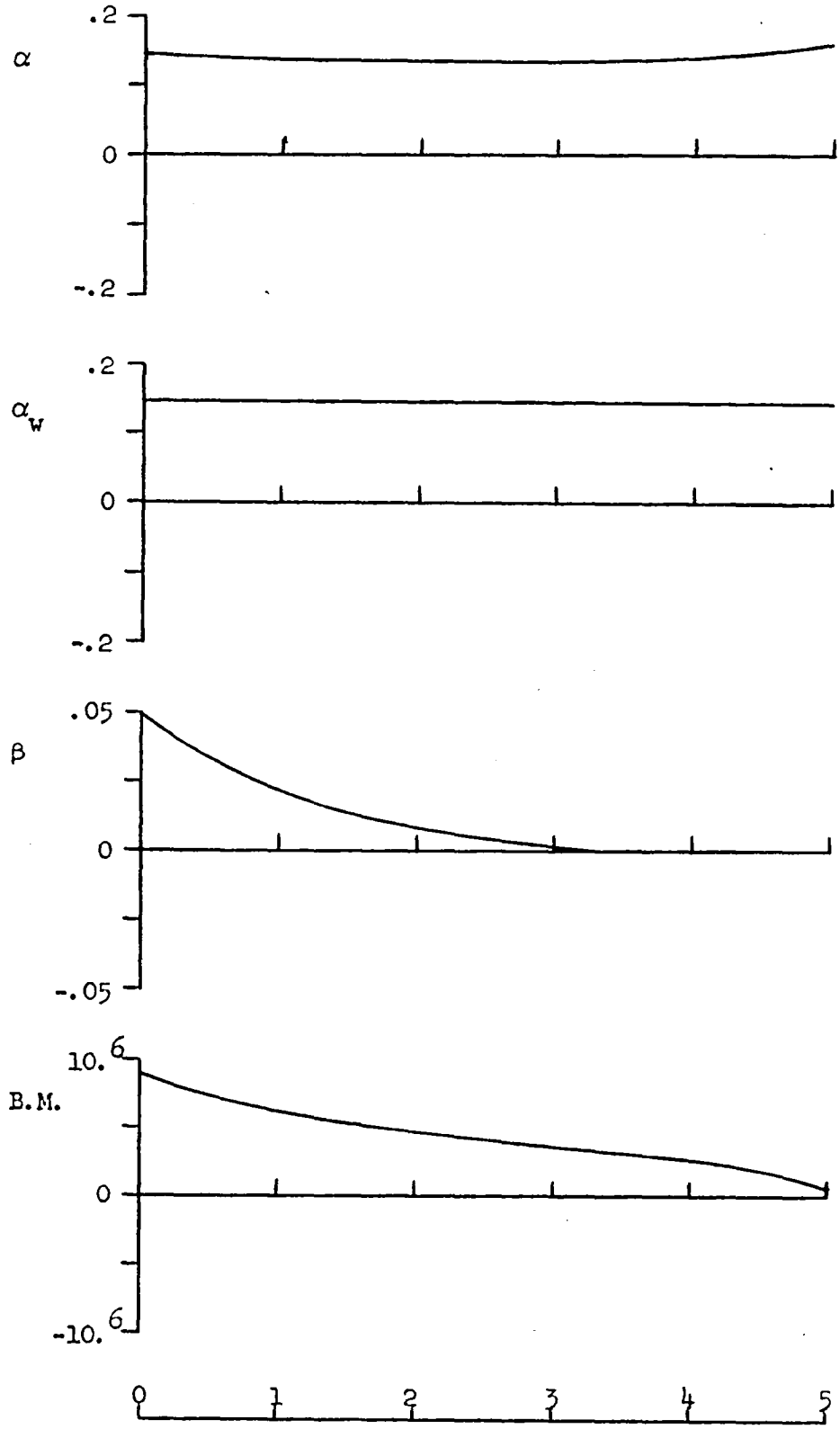


Figure 2.13c

Solution curves for Problem 2.11
Using M.S.F.C. α_w

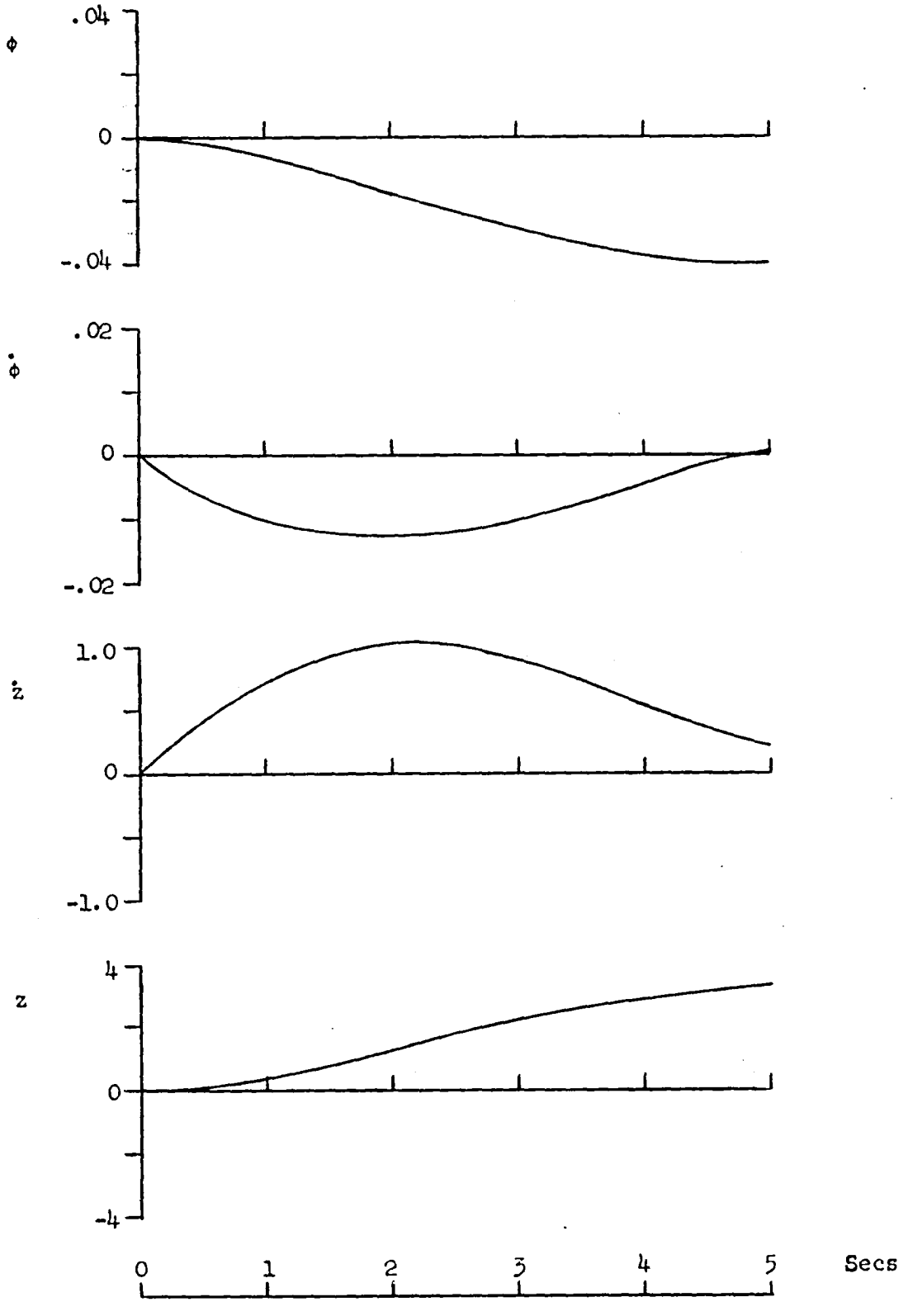
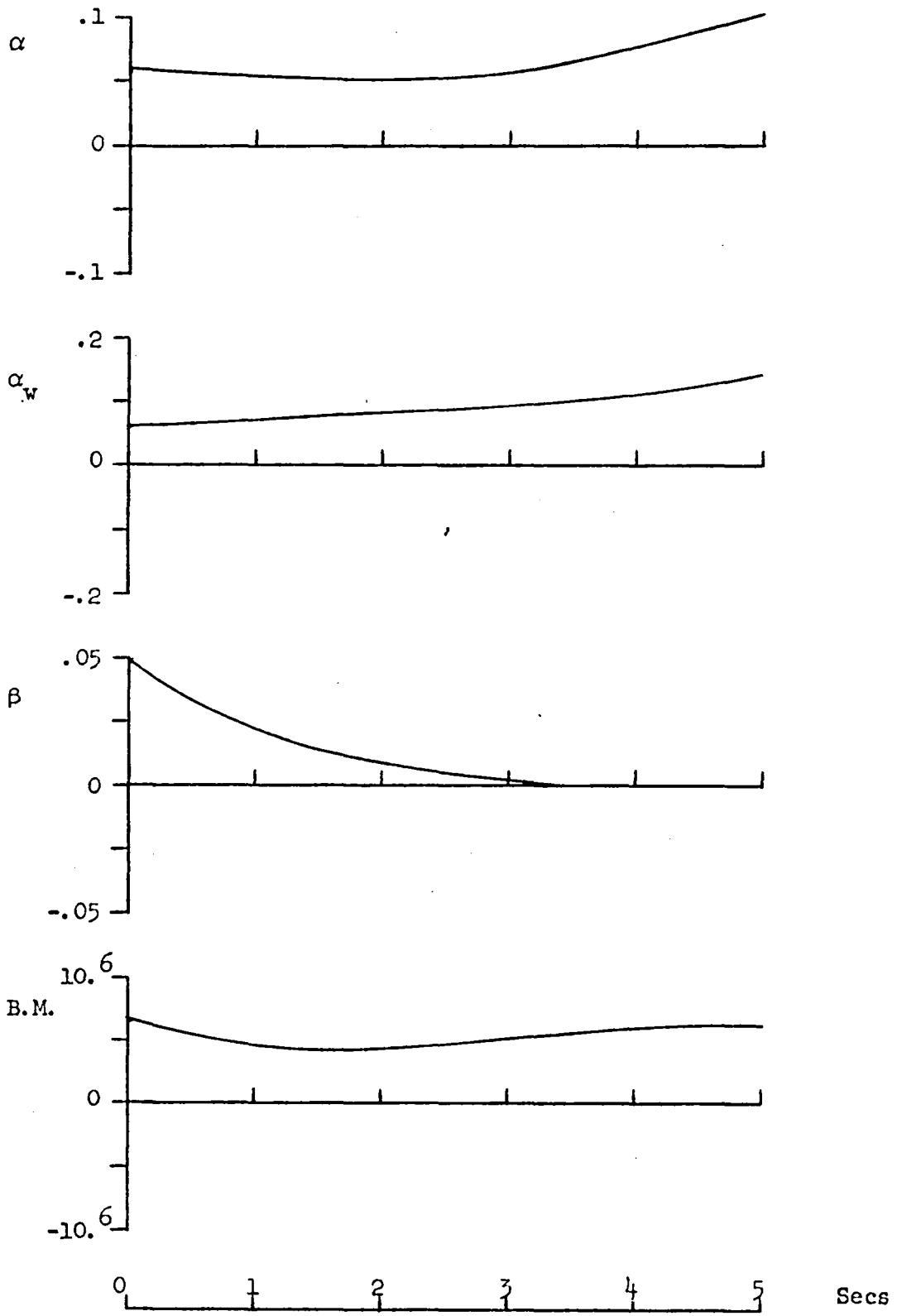


Figure 2.13d

Solution Curves for Problem 2.11
Using M.S.F.C. α_w



$$J = \frac{1}{2}\int_0^5 (x_1^2 + x_2^2 + .004x_3)\, dt$$

The second Gradient Technique (section 2.12) was originally used to solve this problem. There are several different ways to handle fixed terminal conditions such as $\dot{\phi}(T) = 0$ [8], [9]. The penalty function approach in [9] has been used here because it is readily applicable to the specific technique proposed in section 2.12. Using this approach, a penalty function $250x_2^2(T)$ is added to the performance index.

$$J' = J + 250x_2^2(T)$$

Using J' instead of J, the problem is then solved with $x_2(T)$ free. Inspection of the solution curves in Figure 2.14a shows that the peak value of $x_2(t)$ is approximately 0.424 rad/sec (2.4 deg/sec) while $x_2(T) \approx .0021$ rad/sec (.012 deg/sec). Thus, by forcing $x_2(T)$ to be close to zero, the penalty function produced the desired effect.

Since $\beta(t)$ does not appear in the performance index, the Minimax Principle indicates that it should be bang-bang. Inspection of the curve for $\beta(t)$ in Figure 2.14b shows that it is not quite bang-bang. This is characteristic of the Gradient Technique [9]. Had the problem been allowed to run longer (i.e., more iterations), $\beta(t)$ would have looked more like a bang-bang function. At the termination of computations, the change in performance index was less than .1 %. Thus little would have been gained in terms of performance index by running the problem any longer.

This problem, with the same α_w, was also solved using the α_w data provided by M.S.F.C. It was found that the value of the performance index obtained using the M.S.F.C. α_w ($J' = 0.18994$) was <u>larger</u> than the value obtained with the worst α_w ($J' = 0.012202$).

Figure 2.14a

Solution Curves for Problem 2.12
Using Worst α_w

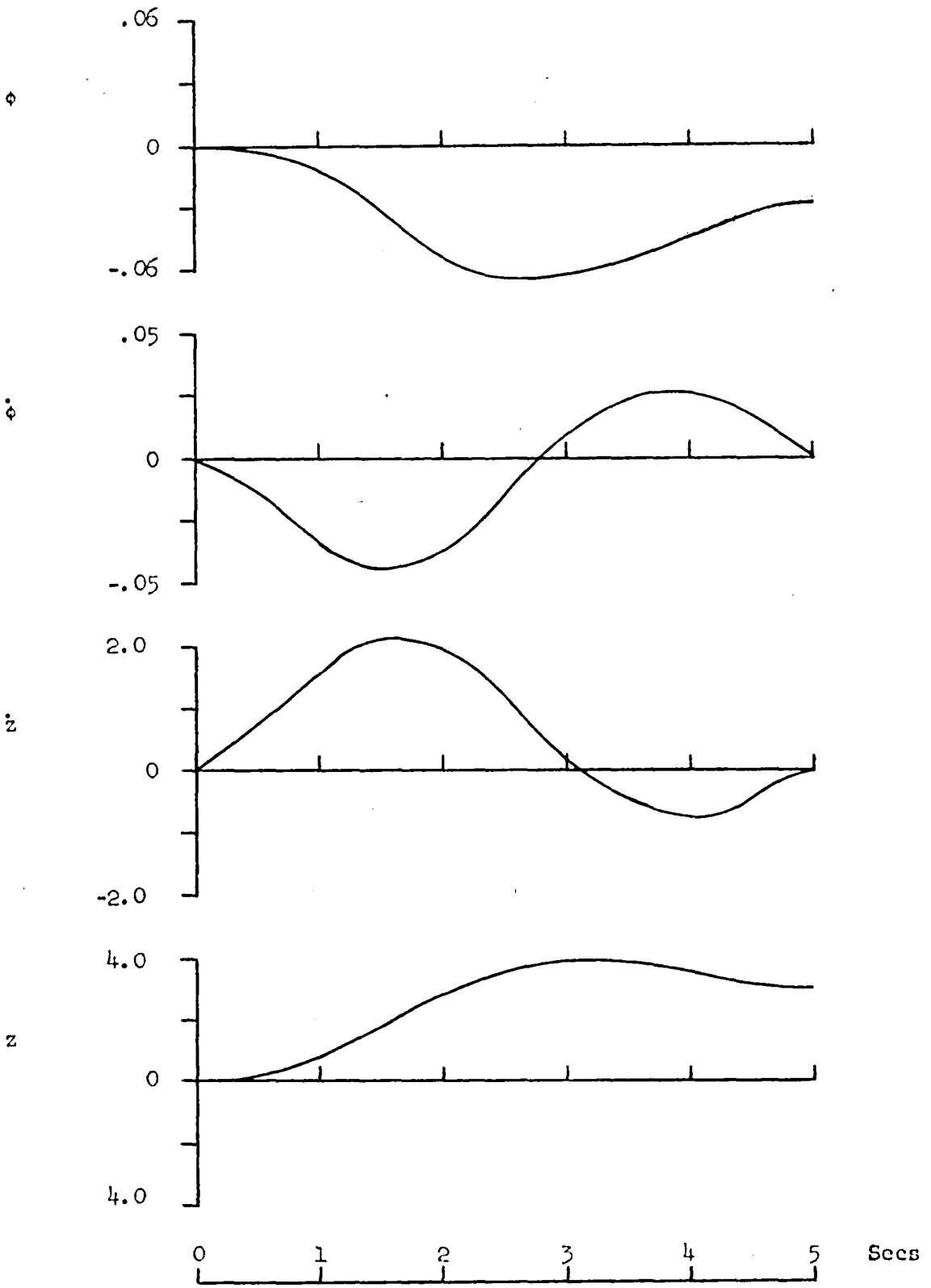


Figure 2.14b

Solution Curves for Problem 2.12
Using Worst α_w

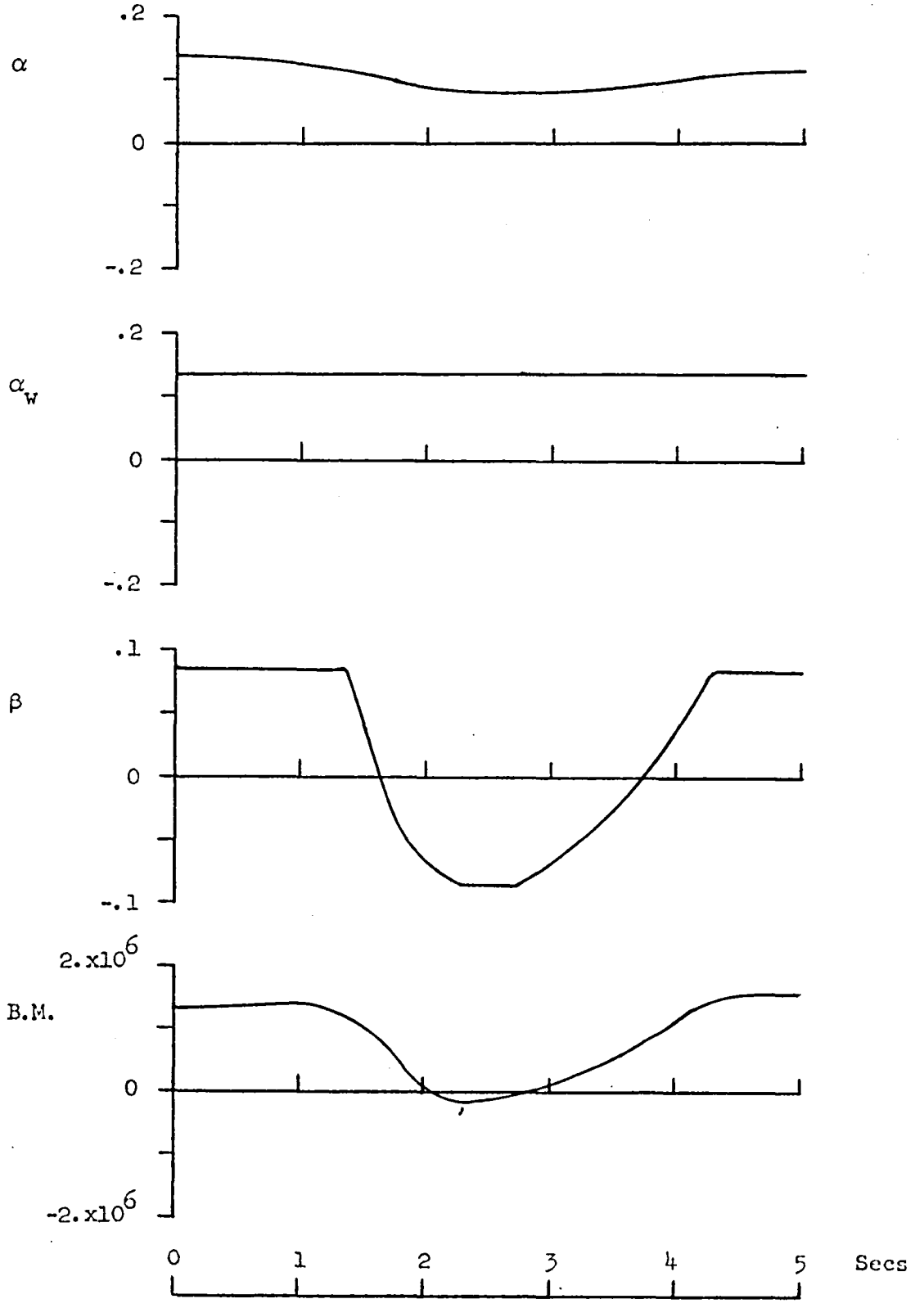


Figure 2.14c

Solution Curves for Problem 2.12
Using M.S.F.C. α_w

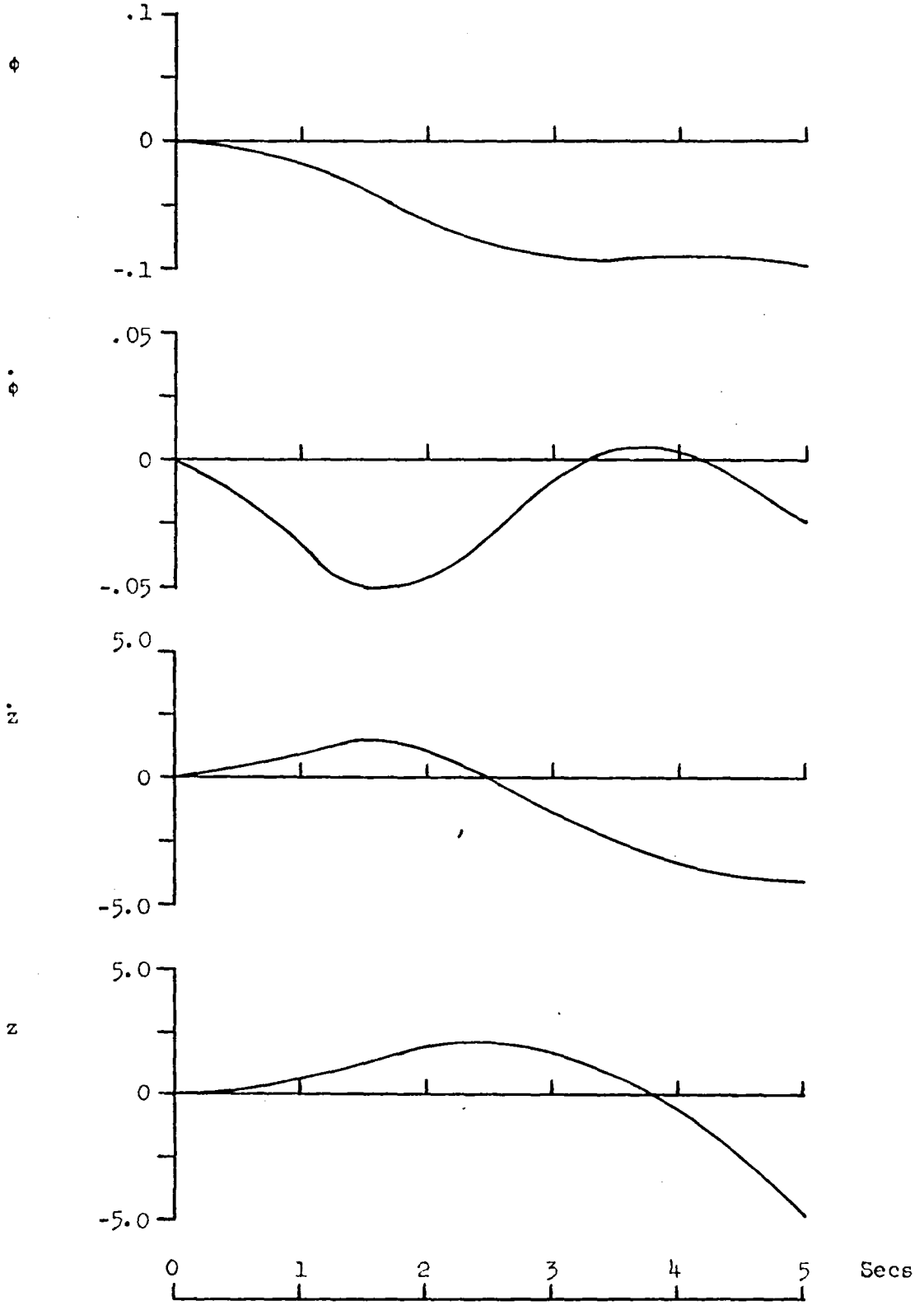
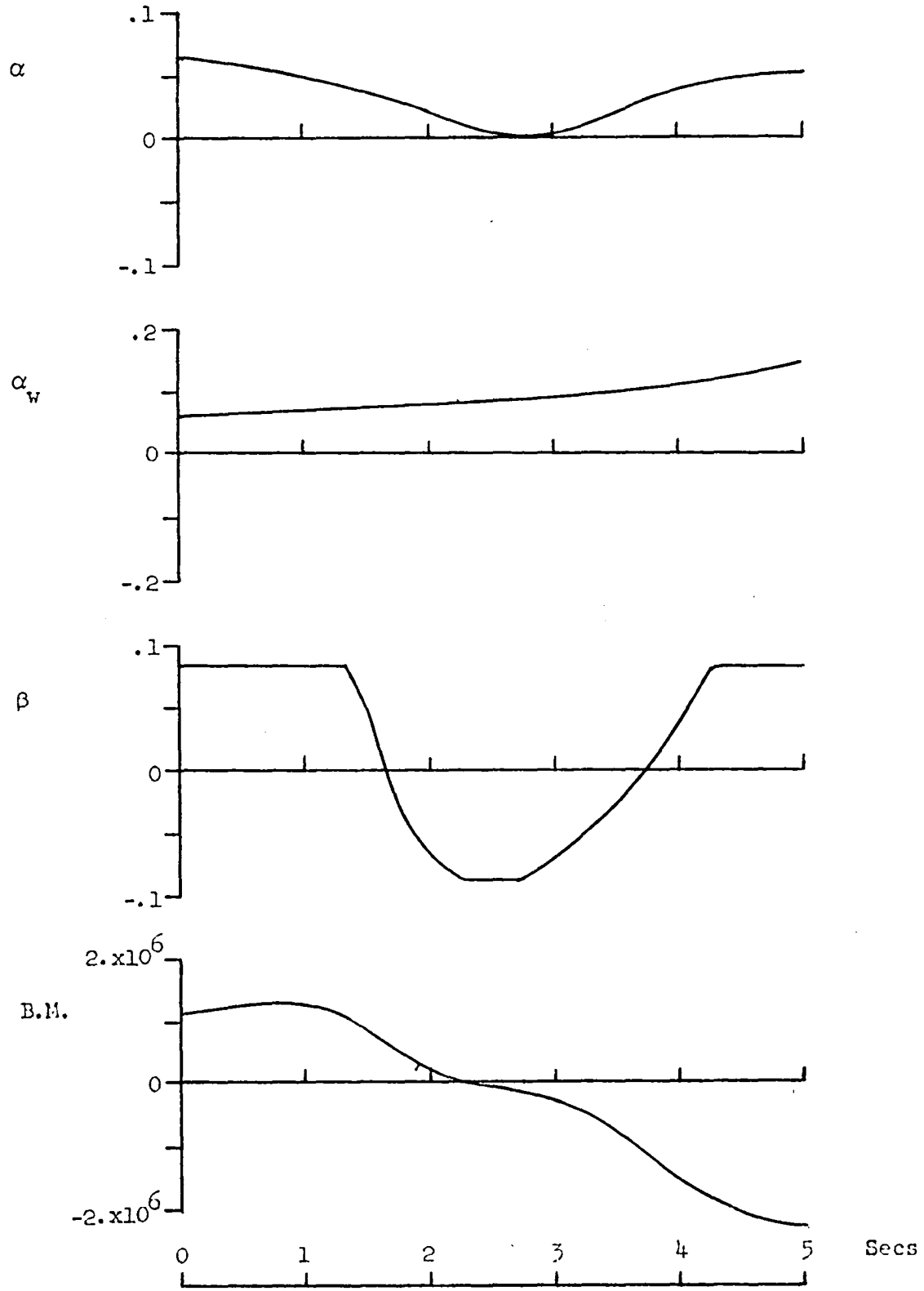


Figure 2.14d

Solution Curves for Problem 2.12
Using M.S.F.C. α_w



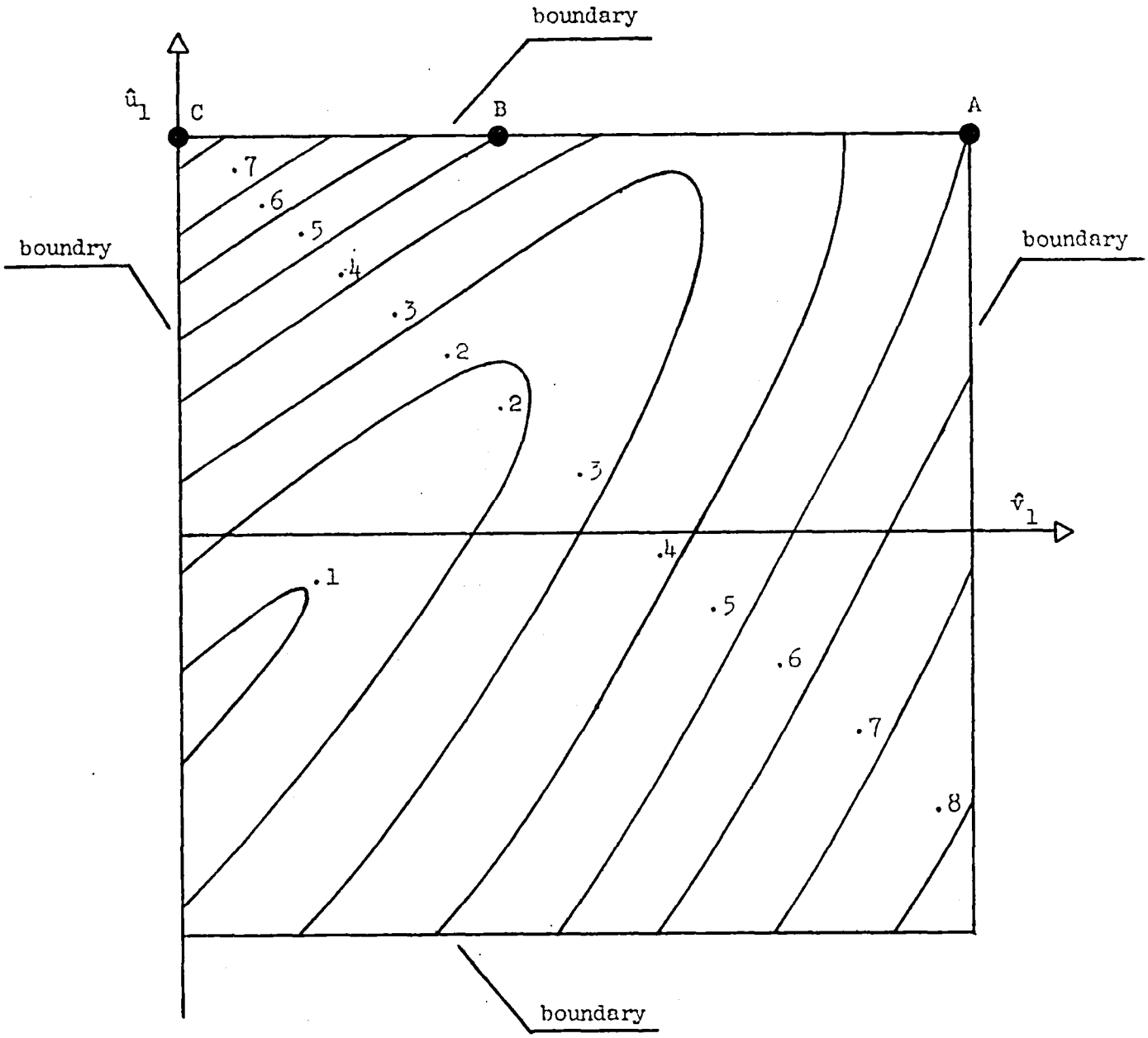


Figure 2.15

Level Curves Corresponding to a One-Dimensional Minimax Problem with Only a Local Solution

The explanation for this unexpected result can best be understood in terms of a one sampling period (N = 1) example. With reference to Figure 2.2 in section 2.12, consider the level curves shown in Figure 2.15. Attention is focused at point A. If $\hat{u}_1$ is changed from its value at A the

performance index increases. If $\hat{v}_1$ is changed slightly from its value at A, the performance index decreases. Thus, point A corresponds to a Minimax solution (at least locally). Note that the Minimax Principle and the Gradient Technique give only local solutions.

Suppose now that $\hat{v}_1$ is changed, holding $\hat{u}_1$ fixed at its value at A. At first the performance index starts to decrease. However, at B the performance index has the same value as at A. Further change in $\hat{v}_1$ (towards C) will cause the performance index to become larger than the value at A. It is reasonable to expect that this is what happened in problem 2.12, of course, in a much higher dimensional space.

It should also be mentioned that this problem was solved again using the M.S.F.C. α_w as the initial guess for the disturbance. The resulting Minimax solution was the same as that obtained when the problem was first solved with an initial guess $\alpha_w(t) \equiv .144$ rad.

Problem 2.13

Given: $x_1 = \phi$, $x_2 = \dot{\phi}$, $x_3 = \dot{z}$

$x_1(0) = .05$, $x_2(0) = x_3(0) = 0$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = .089x_1 - .000171x_3 + .089\alpha_w - .451\beta$$

$$\dot{x}_3 = 26.54x_1 - .0108x_3 + 5.62\alpha_w + 10.94\beta$$

$J = x_1^2(T)$ (see goal 2, section 2.5)

$$|\alpha_w(t)| \leq .1e^{t/3} \quad , \quad |\beta(t)| \leq .1e^{t/3}$$

From the Minimax Principle:

$$H = \lambda_1 x_2 + \lambda_2[.089x_1 - .000171x_3 + .089\alpha_w - .451\beta] + \lambda_3[26.64x_1 - .0108x_3 + 5.62\alpha_w + 10.94\beta]$$

$$\dot{\lambda}_1 = -.089\lambda_2 - 26.64\lambda_3$$
$$\dot{\lambda}_2 = -\lambda_1$$
$$\dot{\lambda}_3 = .000171\lambda_2 + .0108\lambda_3$$

$$\lambda_1(5) = 2x_1(5), \quad \lambda_2(5) = 0, \quad \lambda_3(5) = 0$$
$$\beta^*(t) = .1e^{t/3} \quad \text{sgn}\ (.451\lambda_2 - 10.94\lambda_3)$$
$$\alpha_w^*(t) = .1e^{t/3} \quad \text{sgn}\ (.089\lambda_2 + 5.62\lambda_3)$$

Note: 1. The values of the coefficients used in these plant equations differ slightly from those used in the previous problems. Since these coefficients are only approximate (M.S.F.C. indicates in the introduction of the provided working papers the possibility of $\pm$ 20 % error) the differences do not detract from the quantitive value of results of this problem.

2. Note that the bounds on α_w and β have been chosen as functions of time.

3. As opposed to all previously discussed problems, this problem was solved on a digital-analog hybrid facility. The algorithm used to solve the problem is the one first described in section 2.8.

Since an analog computer was used to solve this problem, a transformation of variables was made to stabilize the differential equations [3]. Specifically

$$\lambda_i' = \lambda e^{-t/3} \qquad i = 1 \text{ to } 3$$
$$x_i' = xe^{-t/3} \qquad i = 1 \text{ to } 3$$

The primed variables were actually used in the analog simulation.

The control $\beta^*(t)$ and the disturbance $\alpha_w^*(t)$ were found to be on their boundarys all the time with no switchings:

$$\beta^*(t) = -.1e^{t/3}$$

$$\alpha_w^*(t) = .1e^{t/3}$$

Values of $\phi(T)$ were obtained using:

1. $\alpha_w = \alpha_w^*$, $\beta = \beta^*$
2. $\alpha_w = 0$, $\beta = \beta^*$
3. $\alpha_w = \alpha_w^*$, $\beta = 0$

Plots of $\phi(t)$ for these three cases are shown in Figure 2.16. Note that $\phi(T)$ for case 1 falls between the values of $\phi(T)$ for the other two cases (2 and 3) as expected.

2.15 Conclusions and Recommendations for Further Study

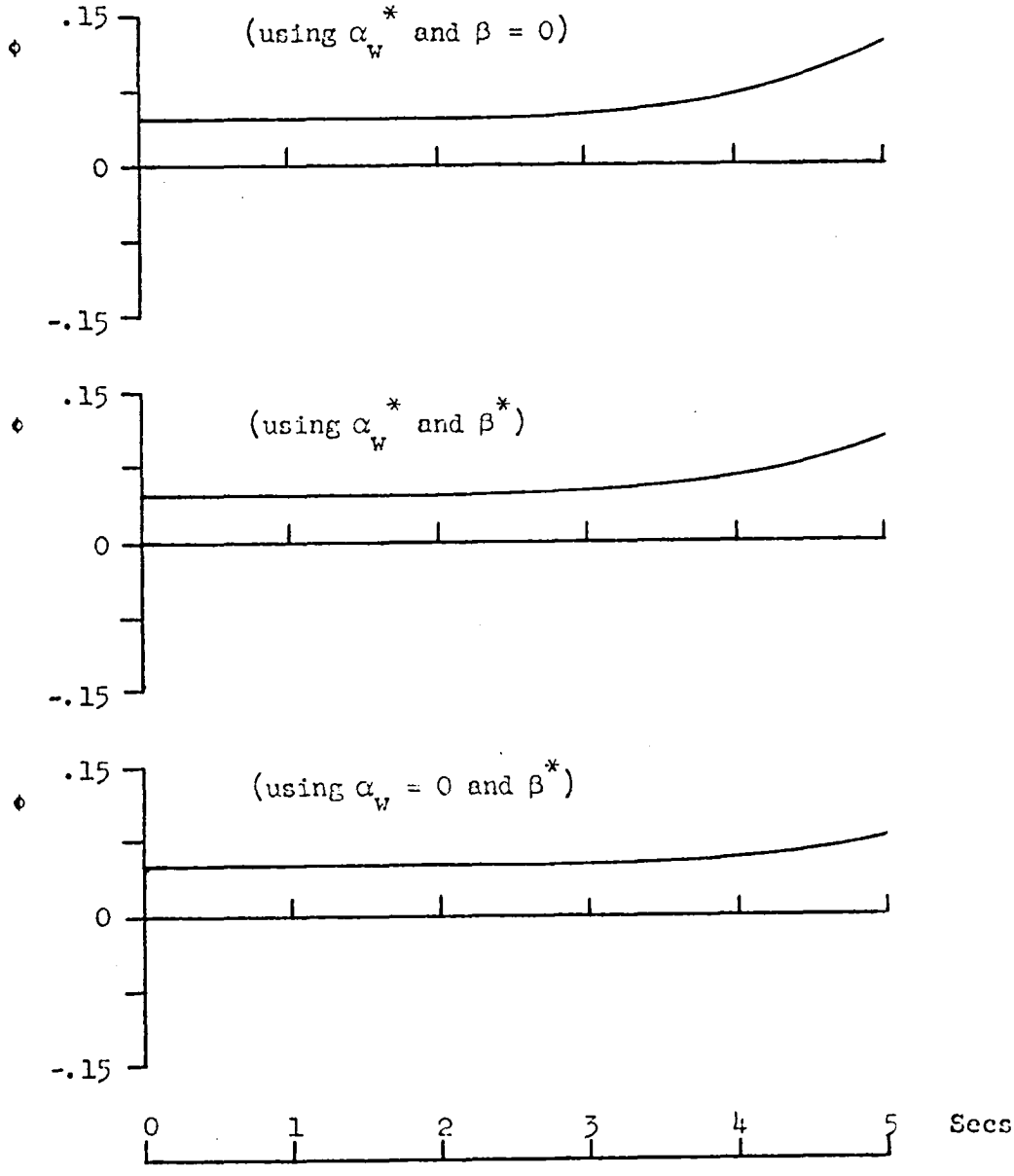
The purpose of this chapter has been to familiarize the reader with the Minimax Problem and with some of the methods for solving it. A formulation of the problem and a discussion of some of its aspects has been presented in sections 2.2 to 2.5.

In stating the Minimax Problem for a physical system, one must relate desired goals to an index of performance which fits within the framework of the problem. Some work has been done on this problem in sections 2.5 and 2.14. Though somewhat obscured by other difficulties associated with the Minimax Problem, the means for choosing a suitable performance index remains as a most important engineering problem.

One approach to the Minimax Problem is the Minimax Principle. To illustrate how the principle is applied, an example has been solved analytically in section 2.7. An important observation resulting from this example is that solutions to the Minimax Problem do not necessarily exist, even for linear plant, quadratic performance index, free terminal state problems.

Figure 2.16

Solution Curves for Problem 2.13



Since the Minimax Principle leads to a two-point boundary value problem, one must generally rely on computational algorithms for solutions. If a solution cannot be obtained, it must be determined whether the computational scheme is not working correctly or whether the specific problem under investigation has no solution. This is a difficult task. It is therefore apparent that there is a need for existence theorems.

An area of research directly related to the Minimax Problem is the boundary value problem. In sections 2.8 to 2.10 three methods for solving boundary value problems are discussed. One of these is Quasilinearization. Several modifications of the original algorithm proposed by Bellman [4] have been presented. These modifications greatly reduce the time and memory requirements on a digital computer. Even with these modifications, the time requirements are still large. Efforts to reduce them further would be of value.

A method for handling sat and sgn functions with Quasilinearization has been suggested and successfully applied to several problems. Little is known about how the approximations employed affect convergence. This is another subject requiring further study.

A second method for solving the Minimax Problem is the Gradient Technique. The method consists of two parts: the calculation of the gradients themselves (section 2.12) and the determination of suitable step sizes (section 2.13). Calculation of the gradients presents no problems, even for nonlinear systems [6]. On the other hand, the determination of suitable step sizes is an unsolved problem. Several approximate methods for choosing step sizes have been proposed in the literature [7], [8], [9], [16], but these are only applicable to minimization or maximization problems. In section 2.13, two methods for generating step sizes for linear plants with quadratic performance indices have

been proposed. Both have been successfully applied to some of the problems in section 2.14. The means for choosing suitable step sizes is considered to be second only in importance to the need for existence theorems.

It has been our experience that the ability of the Gradient Technique to converge does not depend upon initial guesses for control and disturbance. For this reason, the Gradient Technique is regarded as the most promising computational scheme that we have investigated.

In section 2.13, the Gradient Technique is compared with the Ping Pong algorithm when the latter is used with the Minimax Principle. It was found that the two methods are quite similar. As a result of the comparison, it has been proposed that the Ping Pong algorithm be used with variable step sizes generated by the methods in section 2.12. This should prove to be an interesting area for further study.

Before concluding this chapter, some final comments on the Minimax Problem itself are in order.

It is noted that the methods proposed for solving the Minimax Problem yield local solutions. By local is meant solutions which satisfy equation (2.6) (section 2.2) for small changes in u and v from their optimal values. The results in problem 2.12 (section 2.14) indicate that the solution obtained is local. This leads one to suspect that some of the other problem solutions are also local.

As evidenced by the lack of similar results in a minimization problem, conditions which will guarantee global solutions to the Minimax Problem are not readily forthcoming. Furthermore, global solutions may not even exist, as was indicated in the example in section 2.7. Thus the engineering significance of the Minimax Problem appears to be significantly less than the practical value of local solutions in an analogous minimization problem.

Let us illustrate the comparative usefulness of local solutions in Minimax Problems by comparing a hypothetical Minimax Problem and a corresponding minimum problem:

Minimum Problem	Minimax Problem
$J(u_1^*) = 10$	$J(u_1^*, v_1^*) = 10$
$J(u_2^*) = 5$	$J(u_2^*, v_2^*) = 5$
$J(u_3^*) = 1$	$J(u_3^*, v_3^*) = 1$

For the minimum problem, assume that u_1^*, u_2^* and u_3^* are three local solutions. For the Minimax Problem (u_1^*, v_1^*), (u_2^*, v_2^*) and (u_3^*, v_3^*) are also local solutions.

Consider first the minimum problem and assume that only u_1^* and u_2^* have been found. Without knowledge of the existence of u_3^*, one would use u_2^* as the control. Two points need to be made. First, there is no difficulty in choosing between u_1^* and u_2^* since $J(u_2^*) < J(u_1^*)$. Second, one is assured that if u_2^* is used, the value of the performance index will be no greater than 5 (indeed it will be precisely 5).

Now consider the Minimax Problem. Again assume that the first two solutions are known. A difficulty arises, however, when one tries to choose between the two solutions. The numerical values of each performance index may be completely irrelevant unless the precise $v^*(t)$ calculated actually prevails. For example, suppose that the solution giving the lower valued performance index is chosen. Assume then that the system is controlled with u_2^* and an arbitrary (but admissible) distrubance $\bar{v}$ is encountered. It is quite possible that $J(u_1^*, \bar{v}) < J(u_2^*, \bar{v})$ thus implying that better control could have been achieved using u_1^*. Of course one can give a similar argument if u_1^* is used as the control instead of u_2^*. Thus, there is no intelligent way of choosing between two local Minimax solutions. This situation is contrasted with the corresponding situation

in the minimum problem (the first point) which can easily be resolved.

This difficulty is viewed as minor compared to the following problem. It is possible that $J(u_2^*, \bar{v})$ will be greater than 5. That is, an upper bound on the value of the performance index using u_2^* can only be claimed for $\bar{v}$ sufficiently close to v_2^*. Unfortunately, there is no control over $\bar{v}$. Thus for some $\bar{v}$, the performance index could be huge causing a possible catastrophic failure in the system. For this reason, local Minimax solutions must be supplemented by additional physical considerations to be meaningful in an engineering sense.

Let us summarize the important points made above:

1. The presently available methods for solving the Minimax Problem provide only local solutions. The possibility of finding solutions which are indeed only local has been illustrated in Problem 2.12. The impossibility of finding global solutions for some problems is suggested by the results of the example in section 2.7.
2. The means for checking for global solutions is not available and with some reflection on the corresponding minimum problem, it is expected that such means are not readily forthcoming.
3. Without knowledge of global solution (which may not even exist for some problems), one cannot be sure that the u^* used, in the presence of some $\bar{v}$, will not produce a value of performance index well in excess of the Minimax value, thus creating the possibility of catastrophic failure.

REFERENCES

1. Gibson, J. E., Nonlinear Automatic Control, McGraw-Hill Book Company, New York, 1963, Chapter 11.

2. Newton, G. C., Jr., Gould, L. A., and Kaiser, J. F., Analytic Design of Linear Feedback Controls, John Wiley & Sons, Inc., New York, 1957.

3. Howard, D. R. and Rekasius, Z. V., "Error Analysis with the Maximum Principle," IEEE Transactions on Automatic Control, vol. AC-9, n.3, July, 1964, pp. 223-229.

4. Bellman, R. E., Adaptive Control Processes: A Guided Tour, Princeton University Press, Princeton, New Jersey, 1962, Chapter 12.

5. Bellman, R. E. and Dreyfus, S. E., Applied Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1962, Appendix 3.

6. Bryson, A. E., Carroll, F. J., Mikami, K. and Denham, W. F., "Determination of the Lift or Drag Program that Minimizes Re-entry Heating," Journal of the Aerospace Sciences, vol. 29, n. 4, April, 1962, pp. 420-430.

7. Ho, Y. C. and Brentani, P. B., "On Computing Optimal Control with Inequality Constraints," SIAM Journal on Automatic Control, Ser. A, vol. 1, n.3, 1963, pp. 319-347.

8. Bryson, A. E. and Denham, W. F., "Optimal Programming Problems with Inequality Constraints, II: Solution by Steepest-Ascent," AIAA Journal, vol. 2, n.1, January, 1964, pp. 25-34.

9. Kelley, H. J., "Method of Gradients," Optimization Techniques, edited by G. Leitmann, Academic Press, New York, 1962, Chapter 6.

10. Isaacs, R., Differential Games, John Wiley & Sons, Inc., New York, 1965.

11. Berkovitz, L. D., "A Variational Approach to Differential Games," Advances in Game Theory, Annals of Mathematics Studies, Princeton University Press, Princeton, New Jersey, 1964, pp. 127-174.

12. Berkovitz, L. D., "A Differential Game Without Pure Strategy Solutions on an Open Set," Advances in Game Theory, Annals of Mathematics Studies, Princeton University Press, Princeton, New Jersey, 1964, pp. 175-194.

13. Kahne, S. J., "Notes on Two-Point Boundary Value Problems," IEEE Transactions on Automatic Control, vol. AC-8, n.3, July, 1963, p. 257.

14. Kalaba, R., "On Nonlinear Differential Equations, the Maximum Operation, and Monotone Convergence," Journal of Math. and Mech., vol. 8, 1959, pp. 519-574.

15. Bellman, R., Stability Theory of Differential Equations, McGraw-Hill Book Co., New York, 1953, Chapter 1.

16. Merriam, C. W., III, Optimization Theory and the Design of Feedback Control Systems, McGraw-Hill Book Co., New York, 1964, Chapter 10.

CHAPTER III

THE MINIMAX PRINCIPLE

3.1 Introduction

In Chapter II, the Minimax Problem was stated and two methods of solution were proposed. One of these is the Minimax Principle. The proof of the theorems associated with this principle is the subject of this chapter.

The discussion starts with a statement of the Minimax Problem (in section 3.2) for systems and performance indices which are separable in control and disturbance. This is followed, in section 3.3, by a statement and proof of the Minimax Principle. The principle includes some of the necessary conditions (Theorem 3.3) which must be satisfied by candidate solutions to the Minimax Problem. Similar conditions are stated for the free final-state, fixed terminal-time problem in section 3.4.

3.2 Statement of the Problem

We consider a system described by the following system of differential equations,

$$\frac{dx_i}{dt} = f_i(t,x_1, \ldots, x_n,u) + g_i(t,x_1, \ldots, x_n,v) \qquad (3.1)$$

$$i = 1, 2, \ldots, n,$$

where u is an r-vector called the "control" and v is an s-vector called the "disturbance", and satisfying the initial conditions, $x_i(t_o) = x_i^o$, $i = 1, 2, \ldots, n$. We set

$$J(u, v) = \int_{t_o}^{t_1} [f_o(t, x_1(t), \ldots, x_n(t), u(t)) + g_o(t, x_1(t), \ldots, x_n(t), v(t))]\, dt \qquad (3.2)$$

where f_o and g_o are scalar functions of their arguments. We suppose that $u = u(t)$ belongs to a subset U of all measurable r-vectors, defined on $[t_o, t_1]$ and $v = v(t)$ belongs to a subset V of all measurable s-vectors also defined on $[t_o, t_1]$. We further suppose that

$$f_i,\ g_i,\ f_o,\ g_o\ \frac{\partial f_j}{\partial x_i},\ \frac{\partial g_j}{\partial x_i},\ \frac{\partial f_o}{\partial x_i},\ \frac{\partial g_o}{\partial x_i},\ \frac{\partial^2 f_j}{\partial x_i \partial x_k},\ \frac{\partial^2 g_j}{\partial x_i \partial x_k}$$

$$i,j,k = 1, 2, \ldots, n$$

are continuous on X x U or X x V, where X is the n-dimensional space of the vectors x. Let S be a smooth (n-1)-dimensional manifold in X.

We now pose the following problem. Among all pairs of admissible controls u(t) in U and possible disturbances v(t) in V which transfer the phase point from the position $(x_1^o, x_2^o, \ldots, x_n^o)$ to the manifold S, find a pair (u(t), v(t)) for which the functional J(u, v) assumes the value J_o, where

$$J_o = \min_{u \epsilon U} \max_{v \epsilon V} J(u, v) \qquad (3.3)$$

and $t_1 - t_o$ is the time of transfer from x^o to S. We are assuming, of course, that such a number J_o exists and that there exists a pair (u(t), v(t)) with u in U and v in V which actually does transfer the point $(x_1^o, \ldots, x_n^o)$ to S. We suppose t_o fixed and t_1 variable.

A control $u(t) \epsilon U$ will be called admissible if

(a) U is a subset of bounded measurable r-vectors on

$t_o \leq t \leq t_1$ and $u(t) \epsilon U$

or (b) if u(t) is admissible, $t_o \leq t \leq t_1$, and if μ is an arbitrary point of U, then

$$u_1(t) = \begin{cases} \mu \text{ for } t' \le t \le t'' \\ u(t) \text{ for } t_1 \le t \le t' \ , \ t'' \le t \le t_1 \ , \end{cases}$$

is also admissible.

or (c) if the interval $t_o \le t \le t_1$ is subdivided into a finite number of subintervals on each of which the control $u(t)$ is admissible, then this control is also admissible on the entire interval, $t_o \le t \le t_1$. A control obtained from an admissible control by a translation in time is also admissible.

An analogous definition holds for possible disturbances.

Let

$$x_o = \int_{t_o}^{t} [f_o(t, x, u) + g_o(t, x, v)]\, dt \qquad (3.4)$$

Then $x_o(t_o) = 0$. Henceforth the symbols x, f, g and x^o will represent the augmented vectors,

$$(x_o, x_1, \ldots, x_n) \ , \ (f_o, f_1, \ldots, f_n), \ (g_o, g_1, \ldots, g_n), \text{ and}$$
$$(0, x_1^o, \ldots, x_n^o).$$

A vector triple $(u^*(t), v^*(t), x^*(t))$ satisfying relation (3.3) will be called optimal.

With the aid of the new notation, we may combine equations (3.1) and (3.4) as

$$\frac{dx}{dt} = f(t, x, u) + g(t, x, v) \ . \qquad (3.5)$$

3.3 The Minimax Principle

We shall consider the auxiliary system of equations,

$$\frac{d\psi_i}{dt} = -\sum_{j=0}^{n} \left[\frac{\partial f_j}{\partial x_i} + \frac{\partial g_j}{\partial x_i} \right] \psi_j \quad , \quad i = 0, 1, 2, \ldots, n \qquad (3.6)$$

We shall define the Hamiltonian H by

$$H(t, x, \psi, u, v) = \sum_{i=0}^{n} \psi_i \, [f_i(t, x, u) + g_i(t, x, v)]. \qquad (3.7)$$

With the aid of this Hamiltonian systems (3.5) and (3.6) may be written in canonical form,

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial \psi_i} \quad , \quad i = 0, 1, 2, \ldots, n \qquad (3.8)$$

$$\frac{d\psi_i}{dt} = - \frac{\partial H}{\partial x_i} \quad , \quad i = 0, 1, 2, \ldots, n \qquad (3.9)$$

Equations (3.6) or (3.9) are linear equations in the variables ψ_i.

For fixed (constant) values of t, ψ, x and v, let

$$m(t, x, \psi, v) = \min_{u \in U} H(t, x, \psi, u, v) \, ,$$

For fixed (constant) values of t, ψ, x, and u, let

$$M(t, x, \psi, u) = \max_{v \in V} H(t, x, \psi, u, v) \, .$$

For fixed values of t, x, ψ let

$$H^*(t, x, \psi) = \min_{u \in U} \max_{v \in V} H(t, x, \psi, u, v)$$

and

$$H^{**}(t, x, \psi) = \max_{v \in V} \min_{u \in U} H(t, x, \psi, u, v) \, .$$

Referring to the proof of the Pontryagin Maximum Principle, [1], if we restrict K_τ, the "cone of attainability," to lie in the positive half-space $\sum_{i=0}^{n} a_i x_i \geq 0$ rather than in the negative half-space, Pontryagin's Maximum Principle becomes the Minimum Principle:

Theorem 3.1. Let $u(t)$, $t_o \leq t \leq t_1$, be an admissible control and let $v(t)$, $t_o \leq t \leq t_1$ be a possible disturbance such that the corresponding

trajectory $x(t)$ of system (3.8) which starts at $x^o = (0, x_1^o, \ldots, x_n^o)$ at time t_o, intersects S at a time t_1. In order that $u(t)$ and $x(t)$ be optimal[1] for a given $v(t)$ it is necessary that there exist a nonzero, continuous vector function $\psi(t) = (\psi_o(t), \psi_1(t), \ldots, \psi_n(t))$ corresponding to the function $u(t)$, $x(t)$ and $v(t)$, in the sense of satisfying (3.9), such that

(i) for all t, $t_o \leq t \leq t_1$, the function $H(t, x(t), \psi(t), u, v(t))$ of the variable u attains its minimum in U at the point $u = u(t)$,

(ii) the relations

$$\psi_o(t) = \text{constant} \geq 0, \qquad m(t_1, x(t_1), \psi(t_1), v(t_1)) = 0 \tag{3.10}$$

are satisfied

(iii) the vector $(\psi_o, \psi_1, \ldots, \psi_n)$ satisfies the transversality condition at $t = t_1$. This condition states that the vector $(\psi_1(t_1), \ldots, \psi_n(t_1))$ is orthogonal to the terminal manifold S at the point $(x_1(t_1), \ldots, x_n(t_1))$.

Since S is (n-1)-dimensional, conditions (ii) and (iii) together provide n independent terminal conditions on the vector $(\psi_o(t), \ldots, \psi_n(t))$. We may normalize these conditions by choosing $\psi_o(t) \equiv 1$. This is permitted by the first of equations (3.10) and by the homogeneous nature of equations (3.7) and (3.9).

Since finding the maximum of a function ϕ is equivalent to finding the minimum of $-\phi$, we also have the following "maximum" principle.

Theorem 3.2. Let $u(t)$ and $v(t)$, $t_o \leq t \leq t_1$ be an admissible control and possible disturbance respectively, such that the corresponding trajectory

---

1) Here optimal is used in the sense of minimizing $J(u, v)$ with respect to u for fixed v.

$x(t)$ of system (3.8) which starts at the point x^o at time t_o intersects S at a time t_1. In order that $v(t)$ and $x(t)$ provide the worst case disturbance and trajectory for a given $u(t)$ it is necessary that there exist a nonzero, continuous vector function $\lambda(t) = \lambda_o(t), \lambda_1(t), \ldots, \lambda_n(t))$ satisfying

$$\frac{d\lambda_i}{dt} = -\frac{\partial H}{\partial x_i} \quad , \quad i = 0, 1, \ldots, n \tag{3.11}$$

and such that,

(i) for all t, $t_o \leq t \leq t_1$ the function $H(t, x(t), \lambda(t), u(t), v)$ attains its maximum in V at the point $v = v(t)$,

(ii) the relations

$$\lambda_o(t) = \text{const.} \geq 0 \, , \tag{3.12}$$

and

$$H(t_1, x(t_1), \lambda(t_1), u(t_1)) = 0$$

are satisfied,

(iii) the vector $(\lambda_o, \lambda_1, \ldots, \lambda_n)$ satisfies the transversality condition at $t = t_1$, which is that the vector $(\lambda_1(t_1), \ldots, \lambda_n(t_1))$ is orthogonal to S at the point $(x_1(t_1), \ldots, x_n(t_1))$.

We shall now apply Theorems 3.1 and 3.2 to the min-max problem. We wish to find $u = u^*(t)$, $v = v^*(t)$ and a corresponding (by equation (3.8)) $x^*(t)$ such that

$$J(u^*, v^*) = \int_{t_o}^{t_1} [f_o(t, x^*(t), u^*(t)) + g_o(t, x^*(t), v^*(t))] \, dt$$

$$= \min_{u \in U} \max_{v \in V} J(u, v).$$

Let us first assume that $u^*(t)$, $v^*(t)$, $x^*(t)$ is indeed a sought solution

to the min-max problem. We shall show that if we seek

$$\max_{v \in V} \min_{u \in U} J(u, v)$$

we actually obtain the same solution, that is

$$\min_{u \in U} \max_{v \in V} J(u, v) = \max_{v \in V} \min_{u \in U} J(u, v) = J(u^*, v^*).$$

Under this condition the problem is referred to as the "Minimax Problem."

From equation (3.7) we obtain,

$$H(t, x, \psi, u, v) = \sum_{i=0}^{n} \psi_i f_i(t, x, u) + \sum_{i=0}^{n} \psi_i g_i(t, x, v).$$

Let

$$H_1(t, x, \psi, u) = \sum_{i=0}^{n} \psi_i f_i(t, x, u),$$

and

$$H_2(t, x, \psi, v) = \sum_{i=0}^{n} \psi_i g_i(t, x, v).$$

Then

$$H(t, x, \psi, u, v) = H_1(t, x, \psi, u) + H_2(t, x, \psi, v).$$

If for given t, x(t) and $\psi(t)$, u(t) is such that

$$H_1(t, x(t), \psi(t), u(t)) = \min_{u \in U} H_1(t, x(t), \psi(t), u)$$

and v(t) is such that

$$H_2(t, x(t), \psi(t), v(t)) = \max_{v \in V} H_2(t, x(t), \psi(t), v)$$

then

$$\begin{aligned} H(t, x(t), \psi(t), u(t), v(t)) &= \min_{u \in U} \max_{v \in V} H(t, x(t), \psi(t), u, v) \\ &= \max_{v \in V} \min_{u \in U} H(t, x(t), \psi(t), u, v) \end{aligned} \tag{3.13}$$

If we fix $u = u^*(t)$, $v = v^*(t)$, $x = x^*(t)$, then from equations (3.9) and (3.11) we see that $\psi = \psi^*(t)$ and $\lambda = \lambda^*(t)$ satisfy the same differential equation. And from the transversality conditions, the second of equations (3.10), the second of equations (3.12) and the remark following the statement of Theorem 3.1, $\psi^*(t)$ and $\lambda^*(t)$ satisfy the same full set of terminal conditions at $t = t_1$. Thus $\psi^*(t) \equiv \lambda^*(t)$ on $t_o \leq t \leq t_1$. Furthermore, if we fix $x^*(t)$, $\psi^*(t)$ and $u^*(t)$ then it follows from Theorem 3.2 that

$$H(t, x^*(t), \psi^*(t), u^*(t), v^*(t)) \leq H(t, x^*(t), \psi^*(t), u, v^*(t)).$$

Thus, $u = u^*(t)$, $v = v^*(t)$, $x = x^*(t)$ is also a solution to the max-min problem, and therefore also to the Minimax Problem. Thus it is clear that an "optimal" pair $(u(t), v(t))$ for the min-max problem is also an "optimal" pair for the max-min problem, and that

$$H^*(t, x, \psi) = H^{**}(t, x, \psi).$$

For fixed $x(t)$, $\psi(t)$, the following relations also hold.

$$m(t, x(t), \psi(t), v(t)) = \min_{u \in U} H_1(t, x(t), \psi(t), u) + H_2(t, x(t), \psi(t), v(t))$$

$$M(t, x(t), \psi(t), u(t)) = H_1(t, x(t), \psi(t), u(t)) + \max H_2(t, x(t), \psi(t), v)$$

We have therefore, the Minimax Principle:

Theorem 3.3. Let $u(t)$, $v(t)$, $t_o \leq t \leq t_1$ be an admissible control and a possible disturbance such that the coresponding trajectory $x(t)$ of equations (3.8) which begins at the point x^o at time t_o is defined on the entire interval $t_o \leq t \leq t_1$, and passes at the time t_1 through a point on the manifold S. In order that $u(t)$, $v(t)$ and $x(t)$ be optimal it is necessary that there exist a nonzero absolutely continuous vector function $\psi(t) = (\psi_o(t), \psi_1(t), \ldots, \psi_n(t))$ corresponding to the functions $u(t)$, $v(t)$, $x(t)$ and satisfying equations (3.9) such that,

(i) for all t, $t_o \leq t \leq t_1$, the function $H(t, x(t), \psi(t), u, v)$ of the variables $u \epsilon U$ and $v \epsilon V$ attains its minimax, viz.,

$$\min_{u \epsilon U} \max_{v \epsilon V} H(t, x(t), \psi(t), u, v)$$

$$= H(t, x(t), \psi(t), u(t), v(t))$$

(ii) $$\min_{u \epsilon U} \max_{v \epsilon V} H(t, x(t), \psi(t), u, v)$$

$$= \max_{v \epsilon V} \min_{u \epsilon U} H(t, x(t), \psi(t), u, v)$$

(iii) $\psi_o(t) = \text{constant} \geq 0$ and

$$H^{**}(t_1, x(t_1), \psi(t_1))$$

$$= H^{*}(t_1, x(t_1), \psi(t_1)) = 0, \qquad (3.14)$$

and if $u^*(t)$, $v^*(t)$, $x^*(t)$, $\psi^*(t)$ are optimal then,

(iv) $$H(t, x^*(t), \psi^*(t), u^*(t), v) \leq H(t, x^*(t), \psi^*(t), u^*(t), v^*(t))$$

$$\leq H(t, x^*(t), \psi^*(t), u, v^*(t)).$$

Theorem 3.3 tells us, in particular, that the Minimax solution is found by solving equations (3.8) and (3.9). (3.8) is solved in forward time with the given initial conditions. (3.9) is solved in backward time with terminal conditions given by the second of equations (3.12) and the transversality condition at the right end. The functions u(t) and v(t) used in these equations are found from

$$H_1(t, x(t), \psi(t), u(t)) = \min_{u \epsilon U} H_1(t, x(t), \psi(t), u)$$

and

$$H_2(t, x(t), \psi(t), v(t)) = \max_{v \epsilon V} H_2(t, x(t), \psi(t), v).$$

3.4 A Theorem for the Fixed Terminal Time Free Right End Problem

A theorem analogous to Theorem 3.3 holds for the case of fixed terminal time and free right end. This is a direct consequence of Theorem 7 of [1].

The Minimax Principle in this case states,

Theorem 3.4. In order that the admissible control $u^*(t)$, the possible disturbance, $v^*(t)$, and the corresponding trajectory $x^*(t)$, $t_o \leq t \leq t_1$, yield a solution of the Minimax Problem with fixed left-hand end point x^o, and a free right-hand end point with times t_o and t_1 given, it is necessary that there exist a nonzero continuous vector function $\psi^*(t) = (\psi_o^*(t), \ldots, \psi_n^*(t))$ corresponding (through equation (3.9)) to the functions $u^*(t)$, $v^*(t)$ and $x^*(t)$ such that,

(i) for all t, $t_o \leq t \leq t_1$, the function $H(t, x^*(t), \psi^*(t), u, v)$ of the variables $u \epsilon U$ and $v \epsilon V$ attains its minimax at the point $u = u^*(t)$, $v = v^*(t)$,

(ii) $\min_{v \epsilon V} \max_{u \epsilon U} H(t, x^*(t), \psi^*(t), u, v)$

(iii) $H(t, x^*(t), \psi^*(t), u^*(t), v) \leq H(t, x^*(t), \psi^*(t), u^*(t), v^*(t)) \leq H(t, x^*(t), \psi^*(t), u, v^*(t))$

(iv) $(\psi_o^*(t_1), \psi_1^*(t_1), \ldots, \psi_n^*(t_1)) = (1, 0, 0, \ldots, 0)$.

It should be remarked here that the Minimax Principle has been derived for the case when one end point of the trajectory is fixed in n-dimensional space and the other end either lies on a manifold of dimension (n-1) or is free. If the terminal manifold is a fixed point or has dimension less than (n-1) then the vectors $\psi(t)$ and $\lambda(t)$ are not a priori the same. In this case one suspects that the Minimax Principle still holds under additional restrictions. If, for instance equations (3.1) are linear or the set

$$A(t,x) = \{y \mid y = f(t, x, u) + g(t, x, v),\ u \epsilon U,\ v \epsilon V\}$$

is convex the Principle may still hold. This problem is currently under investigation.

3.5 Conclusions

In this chapter some of the necessary conditions for the existence of a solution to the Minimax Problem with separable functions have been derived. These conditions, referred to as the "Minimax Principle" were presented in section 3.3. A theorem for the free-terminal state, fixed-terminal time problem has been given in section 3.4.

REFERENCES

1. Pontryagin, Boltyanskii, et.al., The Mathematical Theory of Optimal Processes, Interscience, 1962.

2. Berkovitz, L. D., "A Variational Approach to Differential Games," Advances in Game Theory, Annals of Mathematics Study, Princeton University Press, Princeton, N. J., 1964.

3. Berkovitz, L. D. and Flemming, W. H., "On Differential Games with Integral Payoff," Contributions to the Theory of Games, III, Annals of Mathematics Study No. 39, Princeton University Press, Princeton, N. J., 1957.

4. Berkovitz, L. D., "Variational Methods in Problems of Control and Programming," Journal of Mathematical Analysis and Applications, vol. 3, 1961, pp. 145-169.

5. Flemming, W. H., "A Note on Differential Games of Prescribed Duration," Contributions to the Theory of Games, III, Annals of Mathematics Study, No. 39, Princeton University Press, Princeton, N. J., 1957.

6. Isaacs, R., Differential Games, John Wiley & Sons, Inc., New York, 1965.

CHAPTER IV

APPLICATION OF WORST CASE ERROR ANALYSIS TO THE DESIGN OF A SPECIFIC OPTIMAL CONTROL FOR MODEL VEHICLE NO. 2

4.1 Introduction

In many Minimax Problems of practical importance it is not possible to find an optimal control law, and even when this is possible, the controller can be difficult to instrument. Thus the following question is posed: Is it possible to design a specific controller with general form, fixed a priori, in which one or more parameters are to be adjusted so as to minimize a performance index in the presence of a "worst" disturbance? The form of the specific controller would be chosen for practical convenience. This specific solution could then be compared to solutions obtained for the idealistic case to ensure that no significant degradation in performance would be caused by its use.

In this chapter a procedure is presented for the design of a specific optimal controller for a certain class of systems with disturbances. The method is applied to a linearized, rigid form of Model Vehicle No. 2, in which the parameters of a feedback control law are adjusted to minimize the bending moment at a specific location on the vehicle and at a specific time.

4.2 General Problem Statement

Given a plant

$$\dot{x} = F(t, x, u) + G(t, x, v) \tag{4.1}$$

with initial conditions $x(0) = x_o$

where

$$x = \text{col}\ (x_1, \ldots, x_n)$$

Let u be called the control function and v be called the disturbance. Also assume v(t) is bounded

$$|\, v(t) \,| \leq 1$$

We shall assume that the form of a control law is given with ℓ parameters $a = (a_1, \ldots, a_\ell)$ to be adjusted in an optimal fashion. This is called a specific optimal control problem [1]. Writing $u = h(x,a)$, (4.1) becomes

$$\dot{x} = F(t, x, h(x, a)) + G(t, x, v) \tag{4.2}$$

It remains now to find the value of a that satisfies

$$\underset{a}{\text{Min}} \left\{ \underset{v(t)\epsilon V}{\text{Max}} \int_0^T [f(t, x, h(x,a)) + g(t, x, v)]\, dt \right\} \tag{4.3}$$

$$0 \leq t \leq T$$

The maximization with respect to v(t) is a problem in worst case error analysis [2,3]. The terminal time T is a constant and V is the class of all piecewise continuous functions that satisfy $|\, v(t) \,| \leq 1$.

4.3 Worst Case Error Analysis

Worst case error analysis was analyzed in detail by Howard [2,3] for a class of systems described by equation (4.4)

$$\dot{x} = F(t, x) + b(t)\, r \tag{4.4}$$

where

x - system state vector of n components in the n-dimensional Euclidean state space X.

r - scalar forcing function (system disturbance or input)

t - time

and F and b are real valued vector functions.

The measure of error was taken to be

$$E(T) = g[x(T), T] \tag{4.5}$$

where T was fixed, and g was a function of the terminal time and the state of the system at the terminal time. The forcing function r(t) was bounded between time-dependent bounds:

$$R_1(t) \leq r(t) \leq R_2(t)$$

It was indicated how an admissible r(t) could be found which would produce the largest error in the sense of E(T).

4.4 Specific Control for Systems with Disturbances

Let the system of equations (4.1) be given with a form of the controller with parameters "a" to be adjusted, equation (4.2), and let there be given the performance index, equation (4.3). Hill climbing techniques may be used to search for the best a to minimize the performance index. For a fixed, for example a_1, one may find $v_1^*(t)$, [2], where $v_1^*(t)$ is the worst possible disturbance. For a_2 one may find $v_2^*(t)$. The choice between a_1 and a_2 will depend on the error measured by

$$\underset{\substack{v(t)\epsilon V \\ 0 \leq t \leq T}}{\text{Max}} \int_0^T [f(t, x, h(x,a) + g(t, x, v)]\, dt \tag{4.6}$$

It should be noted that the worst disturbance depends on the controller.

4.5 Example

For the following example, Hill Climbing techniques are used to adjust one parameter of a controller so that the controlled system has the best performance under the worst conditions. This technique involves

computing the worst disturbance whenever a parameter is perturbed in the Hill Climbing procedure. At each step one has to solve a two point boundary value problem. For practical reasons, (computing time), this technique is limited to a small number of parameters to be adjusted. The analysis in this section was done on a digital computer (IBM 1620).

This example is taken from Model Vehicle No. 2. A Drift Minimum control law is assumed for the rigid vehicle. The feedback coefficients are to be chosen to minimize the bending moment at an arbitrary location (90 meters from the bottom end).

The rigid body equations are:

$$\ddot{\phi} + C_1\alpha + C_2\beta = 0 \tag{4.7}$$

$$\frac{F-x}{m}\phi + \frac{N^1}{m}\alpha + \frac{R^1}{m}\beta - \ddot{z} = 0 \tag{4.8}$$

$$\alpha - \phi + \frac{\dot{z}}{V} = \alpha_w \tag{4.9}$$

Assume a specific control law:

$$\beta = a_o\,\phi + a_1\,\dot{\phi} + b_o\,\alpha \tag{4.10}$$

The minimum drift condition is

$$b_o = \frac{\lambda}{C_2\,\frac{F-x}{m}}\,a_o - \frac{C_1}{C_2} \tag{4.11}$$

After some manipulations one gets the following equations

$$\dot{x}_1 = x_2 \tag{4.12}$$

$$\dot{x}_2 = x_1\left[-C_1 - C_2a_o - C_2b_o - \frac{N_2^1}{V}\right] + x_2\left[-C_2a_1 - \frac{N_3^1}{V}\right] + \alpha_w\,(-C_1 - C_2b_o) \tag{4.13}$$

where $x_1 \triangleq \phi$

$x_2 \triangleq \dot{\phi}$

$$N_3^1 = \frac{N^1}{m} + b_o \frac{R^1}{m}$$

$$N_2^1 = (C_2 \frac{N^1}{m} - C_1 \frac{R^1}{m}) a_1$$

Define the performance index to be:

$$PI = \max_{\alpha_w(t)} | F_1 x_1(T) + F_2 x_2(T) + F_3 \alpha_w(T) | \qquad (4.14)$$

The coefficients F_i in equation (4.14) are computed so that the expression

$$F_1 x_1(T) + F_2 x_2(T) + F_3 \alpha_w(T)$$

describes the bending moment.

Now define the Hamiltonian for the system of equations (4.12), (4.13) and the performance index (4.14):

$$H = \lambda_1 x_2 + \lambda_2 [- A_1 x_2 - A_2 x_1 - P_2^1 \alpha_w] \qquad (4.15)$$

where

$$A_1 = C_2 a_1 + \frac{N_3^1}{V}$$

and

$$A_2 = C_1 + C_2 a_o + C_2 b_o + \frac{N_2^1}{V}$$

From equation (4.15) one may write the adjoint equations:

$$\dot{\lambda}_1 = A_2 \lambda_2 \qquad (4.16)$$

and

$$\dot{\lambda}_2 = - \lambda_1 + A_1 \lambda_2 \qquad (4.17)$$

In the system equations (4.12), (4.13) and the adjoint equations (4.16) and (4.17) the coefficients are time dependent. For the following experiments a linear approximation is made for the coefficients for the time interval $60 \leq t \leq 80$ sec. as follows

$$C_1(t') = - 0.01 - 0.0038\ t'$$

$$C_2(t') = 0.4 + 0.0025\ t'$$

$$1/V(t') = 0.032 - 0.0006\ t'$$

$$\frac{N^1}{m}(t') = 4.8 + .038\ t'$$

$$\frac{R^1}{m}(t') = 9.1 + 0.09\ t'$$

$$\frac{F-x}{m}(t') = 17.6 + 0.17\ t'$$

where $t' = (t - 60)$ sec.

The coefficients F_i for equations (4.14) are for $t = 80$ sec. and depend on the feedback coefficients of the controller a_o, a_1 and b_o. The bending moment at $x = 90$ meters and $t = 80$ sec. is given by

$$\text{B.M. } 10^{-6} = \ell_1\alpha + \ell_2\ddot{z} + \ell_3\ddot{\phi} = 2.23\ \alpha - 3.22\ \ddot{z} - 217\ \ddot{\phi}$$

The coefficients F_i are found to be

$$F_1 = \ell_1 + \frac{\ell_1 N_2^1}{VP_2^1} + \frac{\ell_2 N_3^1 A_2}{P_2^1} - \ell_3 A_2$$

$$F_2 = \frac{\ell_1 N_3^1}{VP_2^1} - \frac{\ell_2 N_2^1}{P_2^1} + \frac{\ell_2 N_3^1 A_1}{P_2^1} - \ell_3 A_1$$

$$F_3 = \ell_1 + \ell_2 N_3^1 - \ell_3 P_2^1$$

For the examples given below it is true that the worst error (highest bending moment), would occur at terminal time. For a proof the reader is referred to Howard's thesis [3].

For the worst disturbance the wind assumes its largest value or switches as in Figure 4.4.

The graphs (Figures 4.1, 4.2, 4.3) show the bending moment and switching time of the worst disturbance α_w as a function of the controller parameters for the time varying rigid vehicle. The time intervals considered were 79 to 80 and 75 to 80 seconds.

Figures 4.4, 4.5 and 4.6 show the worst disturbance, states of the system, drift and bending moment as functions of time.

It was assumed that α_w was bounded to be:

$$|\alpha_w| \leq 0.1 \text{ radians.}$$

4.6 Summary

A design procedure has been demonstrated for adjusting a controller parameter to provide optimal worst-case results. One parameter of the feedback law was adjusted in the example of this chapter. The extension to the optimization of two or more parameters appears to be a standard hillclimbing problem.

$T = 1$ sec.
$a_1 = 8.0$

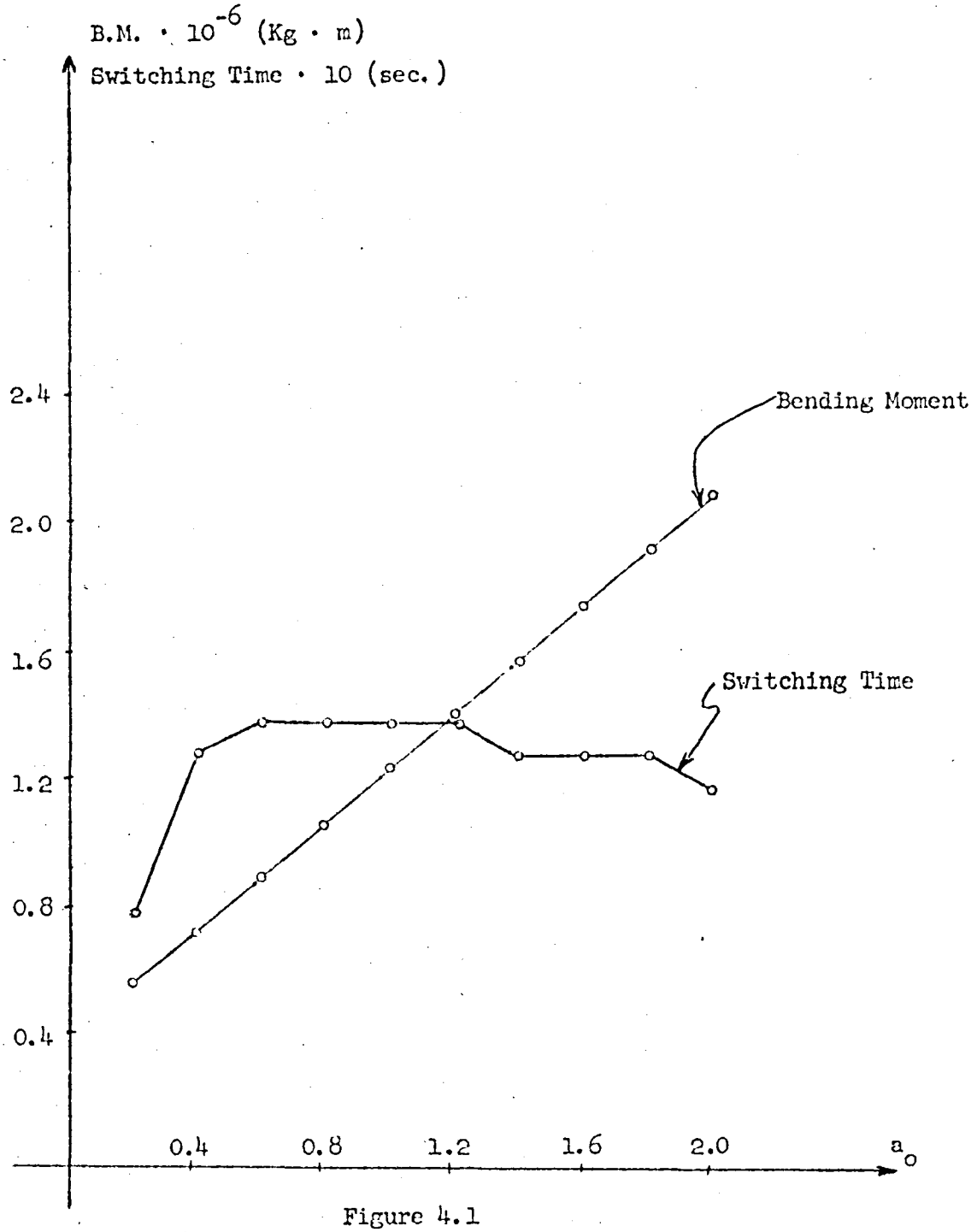


Figure 4.1

Bending Moment and Switching Time, $T = 1$ sec., a_o parameter

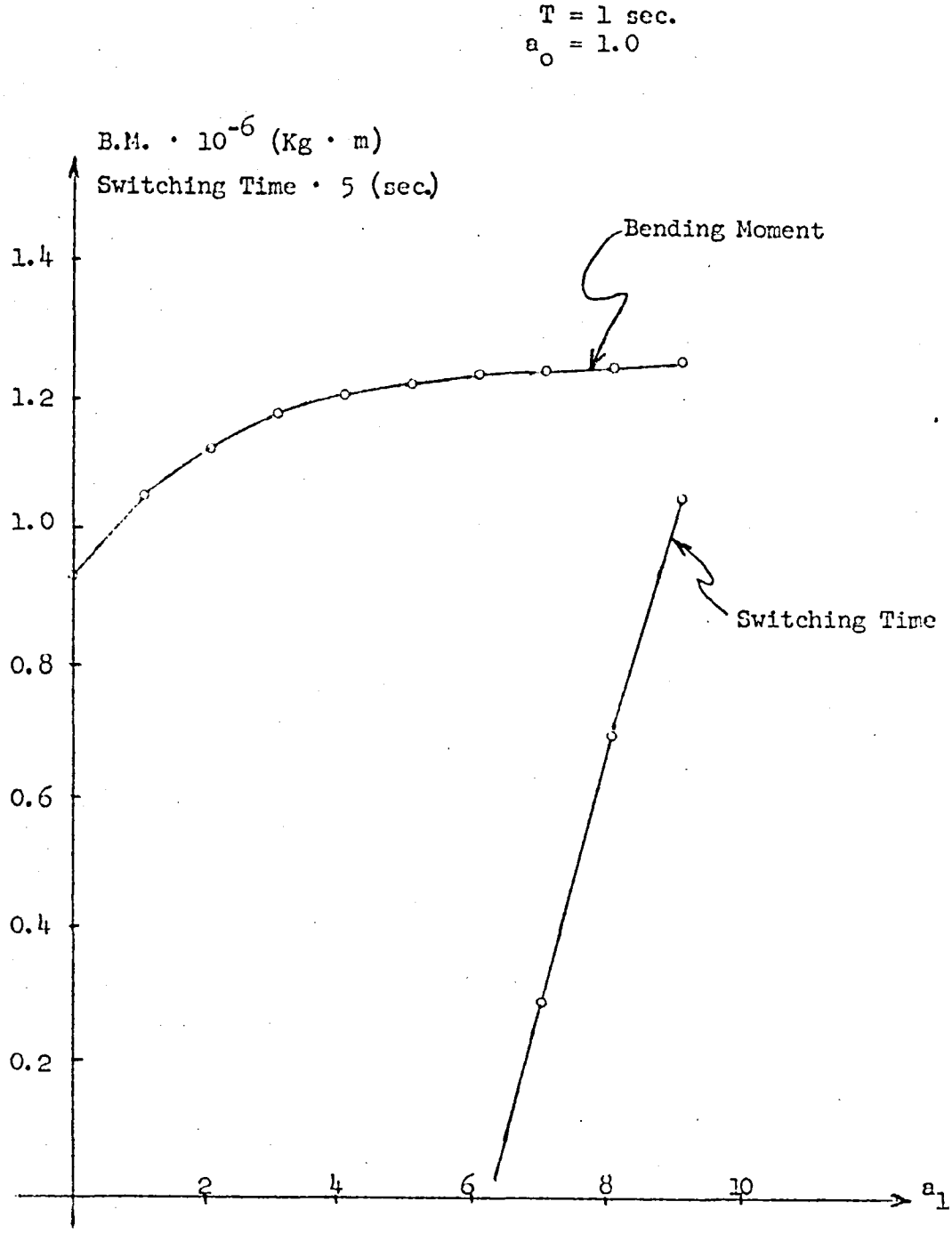


Figure 4.2

Bending Moment and Switching Time, T = 1 sec., a_1 parameter

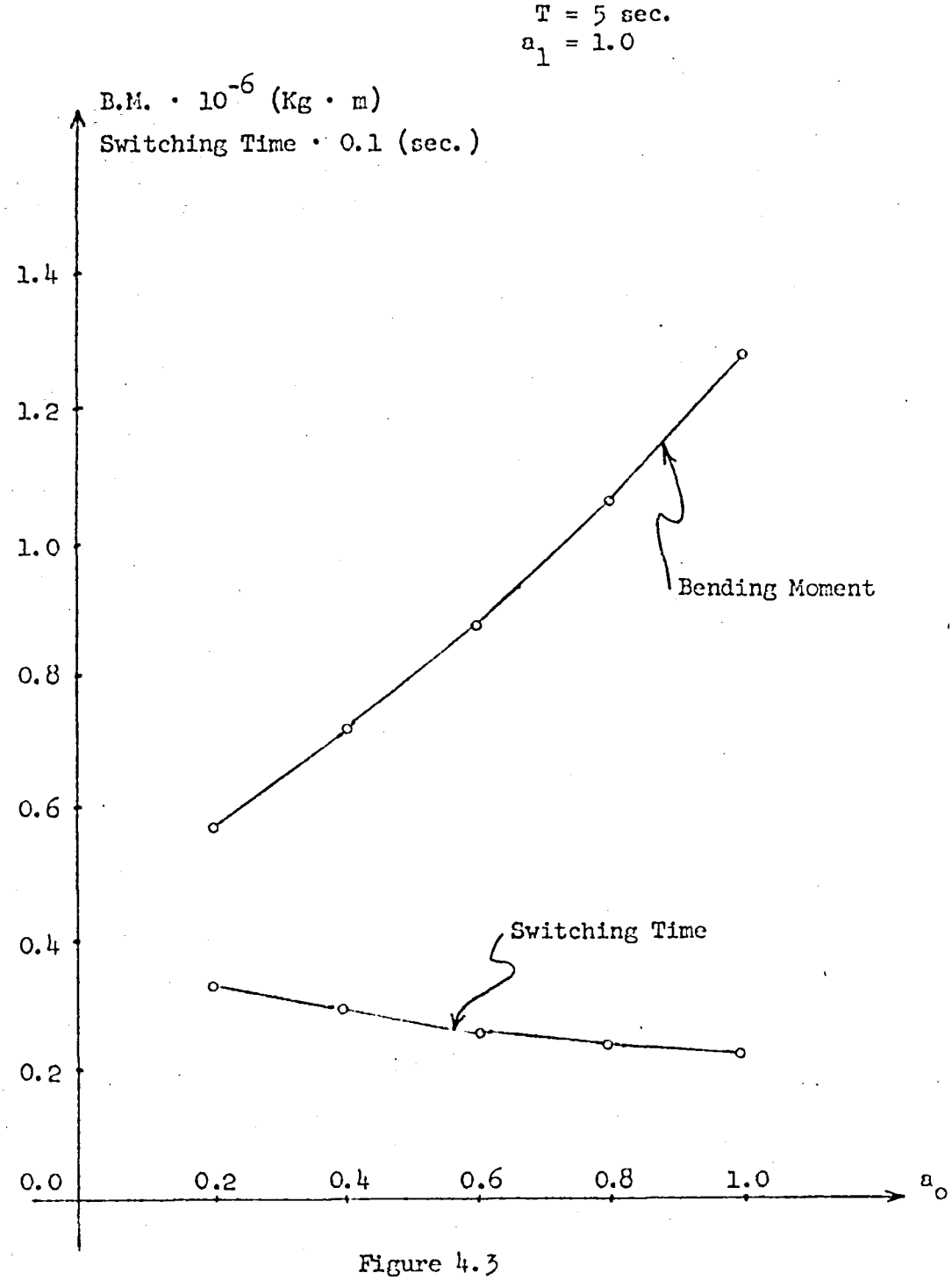


Figure 4.3

Bending Moment and Switching Time, T = 5 sec., a_o parameter

$a_1 = 1.0$
$a_o = 0.2$

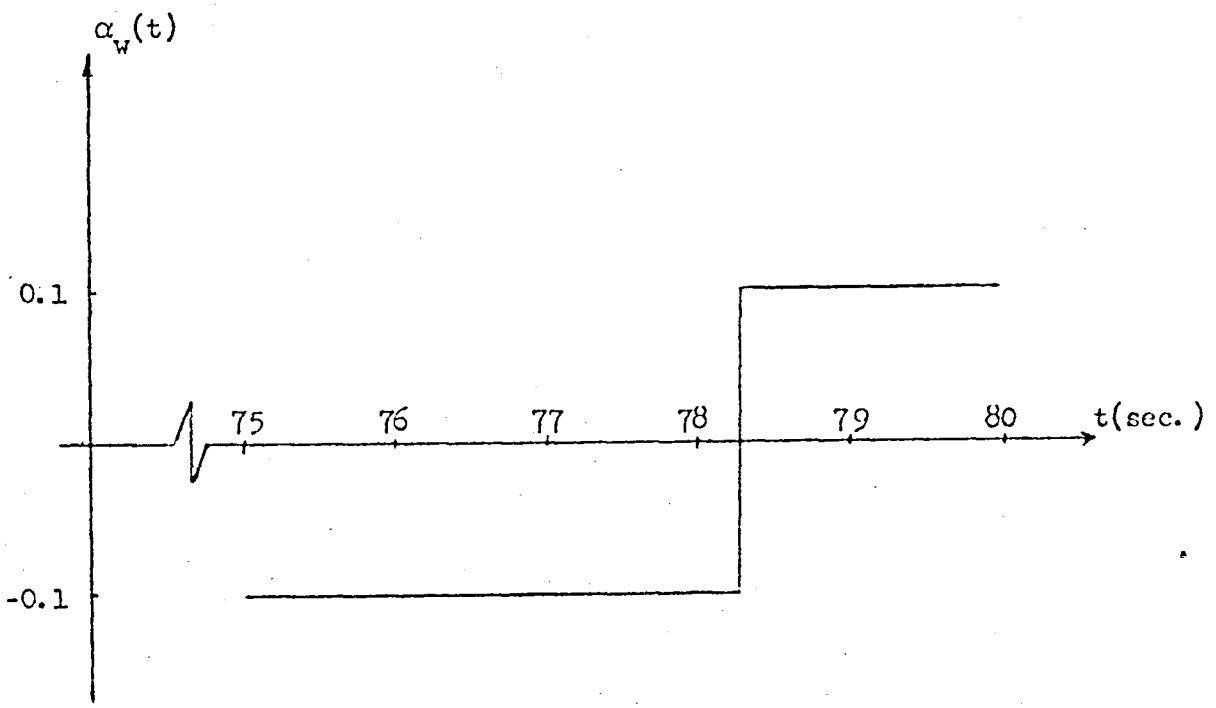


Figure 4.4

Worst Disturbance, T = 5 sec., $a_1 = 1.0$, $a_o = 0.2$

$a_1 = 1.0$ $\quad x_1 \triangleq \phi$

$a_o = 0.2$ $\quad x_2 \triangleq \dot{\phi}$

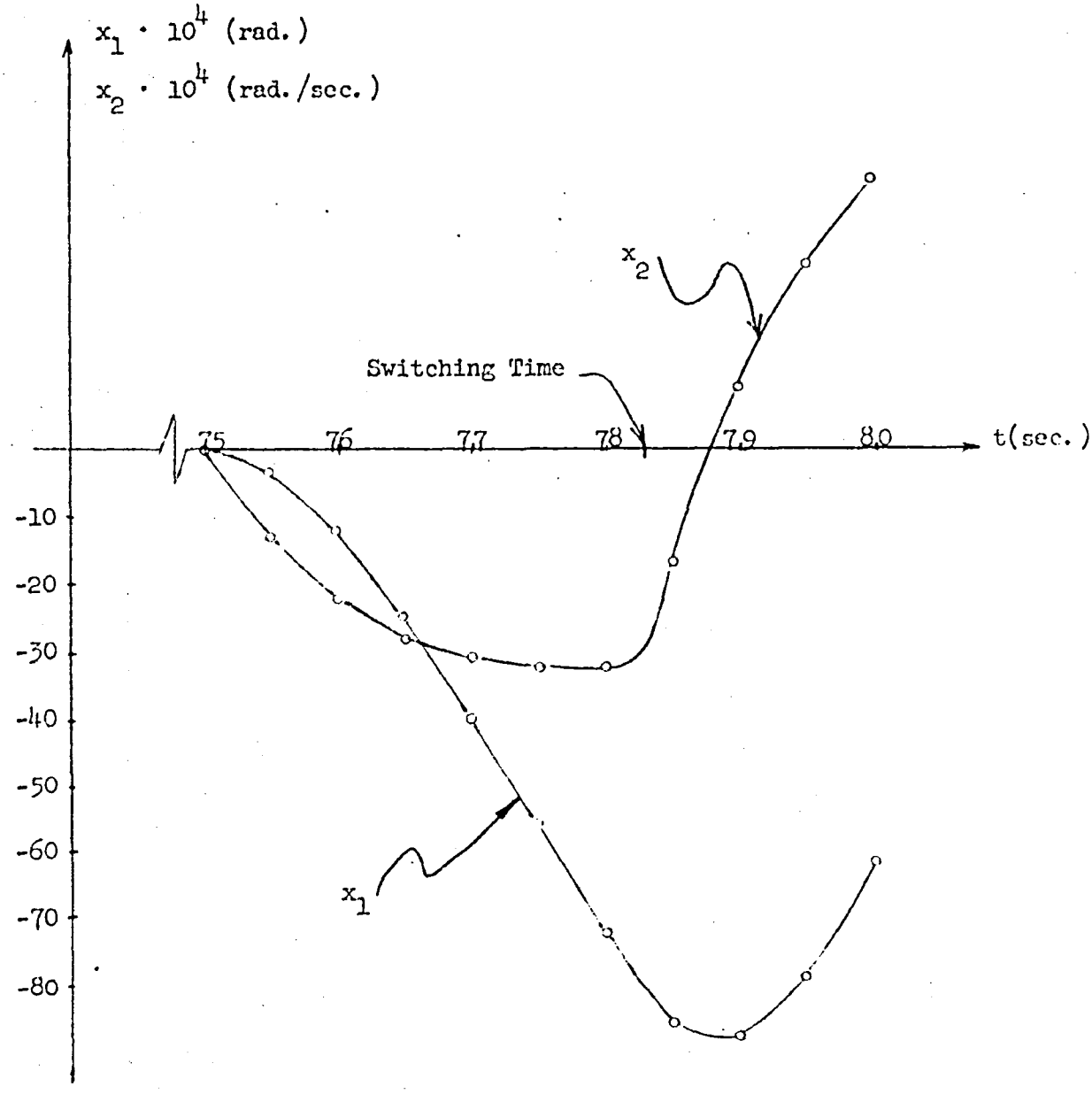


Figure 4.5

System Response to Worst Distrubance, T = 5 sec.

$a_1 = 1.0$
$a_2 = 0.2$

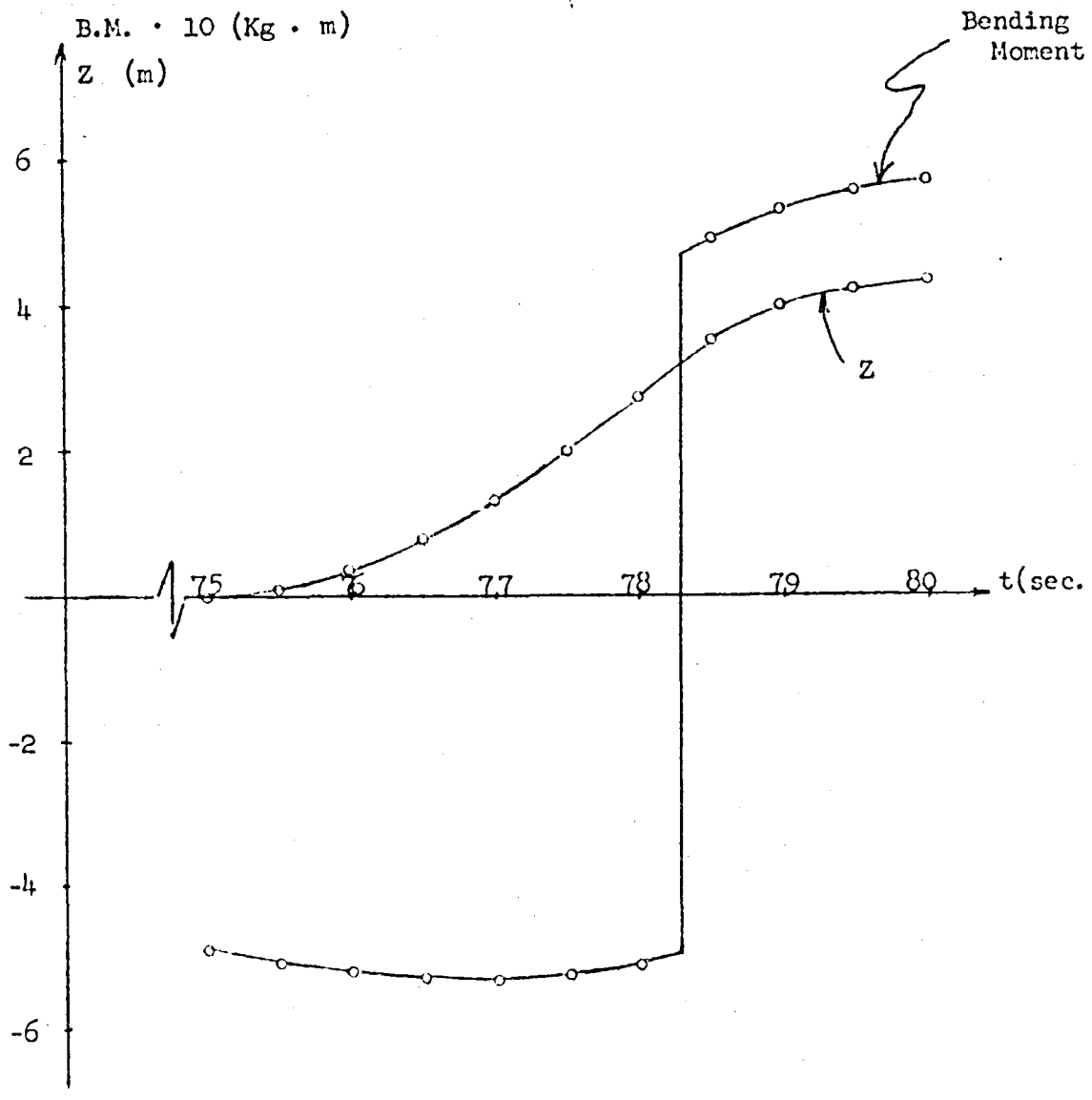


Figure 4.6

Drift and Bending Moment for Worst Disturbance, T = 5 sec.

REFERENCES

1. Agarwal, G. C., "Design of Specific Optimal Control Systems," Ph.D. Thesis, Purdue University, 1965.

2. Howard, D. R., and Rekasius, Z. V., "Error Analysis With the Maximum Principle," IEEE Transactions, PGAC, July, 1964, pp. 223-228.

3. Howard, D. R., "Worst Case Error Analysis Using Optimal Control Theory," Ph.D. Thesis, Purdue University, 1964.

CHAPTER V

APPLICATION OF LINEAR PROGRAMMING TO THE OPTIMAL CONTROL OF MODEL VEHICLE NO. 2

5.1 Introduction

During the last two decades linear programming has grown to a stage where it is finding wide application to problems in industry. This has been made possible because of the availability of high speed digital computers with large memory capacity. Linear programming problems involving hundreds of variables coupled by hundreds of equations can be solved in a fraction of an hour on modern large-scale machines.

In recent years control theorists and engineers have shown increasing interest in the application of linear programming to optimal control problems. Zadeh [1], Whalen [2,3], Porcelli and Fegley [4] and Torng [5] have considered the problem of the design of optimal digital control systems via linear programming. In these references a wide variety of optimal control problems with state and control variable constraints are considered. One of the reasons for the acceptance of linear programming in optimal control theory is the ease with which inequality constraints can be handled.

In this chapter, following an introduction to the theory of linear programming, formulation of a fairly general optimum control problem in the framework of linear programming is presented. In many realistic cases, the solution of such a problem is almost impossible. This is because of the prohibitive number of variables and equations involved even with the powerful computers now available. As a solution to this difficulty, a

new concept of optimal control policy is introduced here. Using this control policy, which is designated as modified optimal control policy, computational solutions to several optimal control problems is presented. Based on the computational results, a comparative study of the behaviour of the optimal control systems under various performances indices is made. The examples on Model Vehicle No. 2 include both time-invariant and time-varying cases. At the end of the chapter concluding remarks on the feasibility of linear programming and the modified optimal control policy are given. The computer algorithms and the subroutines for the solution of the problems discussed in this chapter can be found in appendix C.

5.2 Introduction to Linear Programming

The theory of mathematical programming is concerned with scientific methods for arriving at the best design of a system based on (i) the mathematical model of the system, (ii) the required specifications and (iii) the stated objective. In particular if the system satisfies assumptions of linearity and if the variables involved are non-negative then the corresponding mathematical programming problem is called a linear program and the associated programming technique linear programming (L.P.).

In this section essentials of linear programming are discussed. The optimization problem for Model Vehicle No. 2 will be related to this in a later section.

Basically a problem in linear programming can be stated as follows:

Problem I Minimize

$$z = \underline{c}' \underline{x} \tag{5.1}$$

subject to

$$A \underline{x} = \underline{b} \tag{5.2}$$

$$D\,\underline{x} \leq \underline{e} \tag{5.3}$$

$$\underline{x} \geq 0 \tag{5.4}$$

where

z : Scalar

$\underline{c}$: Constant n_1-vector

$\underline{x}$: Variable n_1-vector

A : Constant $(m_1 \times n_1)$ matrix

$\underline{b}$: Constant m_1-vector

D : Constant $(r \times n_1)$ matrix

$\underline{e}$: Constant r-vector

z is called the cost function, objective form or performance index. Equation (5.2) represents equality constraints on the variables of a process or system (the words 'process' and 'system' are used interchangeably). In other words, equation (5.2) is the mathematical model of the system. Equation (5.3) represents inequality constraints or system specifications. The non-negativity assumption is embodied in equation (5.4).

Equation (5.3) can be subsumed in equation (5.2) by introducing non-negative variables, called slack variables. For example

$$d_{11}\,x_1 + d_{12}\,x_2 \leq e_1$$

can be written as

$$d_{11}\,x_1 + d_{12}\,x_2 + s_1 = e_1 \quad , \quad s_1 \geq 0$$

Here s_1 is called a slack variable. With this modification, Problem I can be reformulated as:

Problem II Minimize

$$z = \hat{\underline{c}}'\,\hat{\underline{x}} \tag{5.5}$$

subject to

$$\hat{A}\,\hat{\underline{x}} = \hat{\underline{b}} \tag{5.6}$$

and $\quad \underline{\hat{x}} \geq 0 \qquad (5.7)$

where

$$\underline{\hat{c}} = \begin{bmatrix} \underline{c} \\ 0 \end{bmatrix} \quad \text{Constant } (n_1+r) \text{ vector}$$

$$\underline{\hat{x}} = \begin{bmatrix} \underline{x} \\ x_{n_1+1} \\ \vdots \\ x_{n_1+r} \end{bmatrix}$$

$x_{n_1+1}, x_{n_1+2}, \cdots, x_{n_1+r}$ are the slack variables.

$$\hat{A} = \begin{bmatrix} A & 0 \\ D & I \end{bmatrix} \quad (m \times n) \text{ matrix}$$

$m = m_1+r$

$n = n_1+r$

$I = (r \times r)$ unit matrix.

The problem posed by equations (5.5), (5.6) and (5.7) is called the standard problem of linear programming. If a vector $\underline{\hat{x}} = \underline{\hat{X}}$ satisfies equations (5.6) and (5.7), then $\underline{\hat{X}}$ is called a feasible solution. If $\underline{\hat{X}}$ is such that not more than m (number of rows in $\hat{A}$) of its components X_j are positive, then it is called a basic feasible solution. A non-degenerate basic feasible solution is a basic feasible solution with exactly m positive components X_j. If a feasible solution $\underline{\hat{X}}$ is such that the right hand side of equation (5.5) is minimized, then it is referred to as minimum feasible solution.

If a feasible solution exists to the Problem II and if the rows of of $\hat{A}$ are independent, then, by elementary row operations and possibly relabeling the variables (5.6) can be written as

$$\begin{bmatrix} 1 & & & 0 & \bar{a}_{1,m+1} & \bar{a}_{1,m+2} & \cdots & \bar{a}_{1,n} \\ & 1 & & & \bar{a}_{2,m+1} & \bar{a}_{2,m+2} & & \bar{a}_{2,n} \\ & & \ddots & & \vdots & \vdots & & \vdots \\ 0 & & & 1 & \bar{a}_{m,m+1} & \bar{a}_{m,m+2} & & \bar{a}_{m,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \vdots \\ \bar{b}_m \end{bmatrix}$$

$$\bar{b}_i \geq 0 \quad , \quad i = 1, 2, \ldots, m \, . \tag{5.8}$$

Equation (5.8) represents a canonical system. The variables $x_1, x_2, \ldots, x_m$ are called basic variables and are non-negative. The rest of the variables, i.e., $x_{m+1}, x_{m+2}, \ldots, x_n$ are non-basic variables and are all zero.

In the foregoing discussion it is tacitly assumed that in $\hat{A}$ the number of rows m is less than the number of columns n. However if $m > n$, it is possible to write an equivalent dual problem which has m columns and n rows. Thus there is no loss of generality in assuming that the number of rows is less than the number of columns. The assumption $n > m$ is always true in the type of problems considered in this report.

A linear programming problem with the coefficient matrix $\hat{A}$ in canonical form, and the right hand side vector $\bar{b}_i \geq 0$, is said to be in feasible canonical form.

5.3 Solution of Linear Programming Problems

A linear programming problem in two variables can be solved graphically, but if the number of variables is more than two, graphical solutions are impossible. In such cases, using the basic theory of linear algebra, it is possible to get a set of basic feasible solutions (assuming a feasible solution exists) by performing a series of elementary row operations. This, however, is a brute force method. In order to find the minimum basic feasible solution, the entire set of basic feasible solution must be

determined. Dantzig's Simplex Method is a systematic way of finding a minimum basic feasible solution for a problem given in the standard form. The Simplex Method is discussed in detail in [6]. Standard computer subroutines are available for the solution of L.P. problems in the standard form.

5.3.1 Variables Not Restricted in Sign

If the variables involved in a problem are not non-negative, then each of these variables must be replaced by the difference of two non-negative variables, which is always possible. Thus if x_j is not restricted in sign, replace it by $(x_j' - x_j'')$ where $x_j' \geq 0$ and $x_j'' \geq 0$. Note that this substitution doubles the number of variables in the standard form.

5.3.2 Inequality Constraints on the Magnitude of a Variable or a Linear Function of the Variables

A constraint of the type

$$| d_1x_1 + d_2x_2 | \leq e_1$$

can be decomposed into two ordinary inequality constraints

$$\text{(i)} \quad d_1x_1 + d_2x_2 \leq e_1$$

$$\text{(ii)} \quad -d_1x_1 - d_2x_2 \leq e_1$$

And these can be handled without any trouble as discussed earlier.

5.4 Optimal Control Theory and Linear Programming

In this section several problems in optimal control theory are formulated in the standard form of linear programming. It is assumed that the plant can be described by a set of linear difference equations. Various types of performance indices are considered. Feasibility of these performance indices, especially from a computational point of view is discussed.

5.4.1 Mathematical Model

Consider a plant described by

$$\underline{x}(K+1) = A(K)\,\underline{x}(K) + h(K)\,u(K) + d(K) \qquad (5.9)$$

where

$$\underline{x}(K) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{state vector}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & & a_{nn} \end{bmatrix} \quad \text{state coefficient matrix}$$

$$h = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix} \quad \text{control coefficient vector}$$

$$d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} \quad \text{Disturbance vector}$$

u = Control variable

K = Number of the sampling interval

Here the control variable is assumed to be a scalar; generalization to the vector case is straight forward.

5.4.2 Design Specification

Let the design specifications (or constraints) be given by

$$B(K+1)\,\underline{X}(K+1) \leq \underline{\alpha}(K+1) \tag{5.10}$$

where

$$B = \begin{bmatrix} b_{11} & b_{12} \cdots b_{1n} & b_{1,n+1} \\ b_{21} & b_{22} \cdots b_{2n} & b_{2,n+1} \\ \vdots & \vdots \quad\quad \vdots & \vdots \\ b_{r1} & b_{r2} \quad\quad b_{rn} & b_{r,n+1} \end{bmatrix}$$

$$\underline{X}(K+1) = \begin{bmatrix} \underline{x}(K+1) \\ u(K) \end{bmatrix}$$

$$\underline{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_r \end{bmatrix}$$

Note that the constraints are also assumed to be linear.

5.4.3 Performance Index

In order for an optimization problem (other than a minimum time problem) to be meaningful, the objective function or the performance index must be a convex function of the state and control variables. A quadratic performance index, which is so common in optimal control theory based on the calculus of variations or the maximum principle, is not directly amenable to linear programming. An equally valid convex performance index is the absolute value type, which, even though not a linear function of the state and control variables, can be handled very easily in linear programming. Equation (5.11) represents a fairly general type of such a performance index.

$$\text{P.I.} = \min_{\substack{u(K) \\ K=0,1,\ldots,N-1}} \sum_{K=0}^{N-1} \left[\underline{c}'(K+1) \,|\, \underline{x}(K+1) \,| + \gamma(K) \,|\, u(K) \,| \right] \qquad (5.11)$$

$\underline{c}' = [c_1, c_2, \ldots, c_n]$

γ = Scalar

N = Number of sampling periods under consideration

$\underline{c}' |\, \underline{x} \,| = c_1 |\, x_1 \,| + c_2 |\, x_2 \,| + \ldots + c_n |\, x_n \,|$

In order to demonstrate how an absolute value type performance index is manipulated, consider, for simplicity, the one-interval control problem.[1] This will also help in formulation of linear programming problem in standard form. Later the formulation will be generalized.

Problem III One-interval Control Problem

Given the plant and the specifications described by equations (5.9) and (5.10) respectively, determine u(K) such that

$$\underline{c}'(K+1) \,|\, \underline{x}(K+1) \,| + \gamma(K) \,|\, u(K) \,| \qquad (5.12)$$

is minimum. Assume that the present state $\underline{x}(K)$ is known.

Since $\underline{x}$ and u are not restricted in sign, replace each $x_j (j=1,2,\ldots,n)$ and u by a difference of two non-negative variables.

Let $$u(K) = u^+(K) - u^-(K)$$

$$x_j(K+1) = x_j^+(K+1) - x_j^-(K+1) \qquad j = 1, 2, \ldots, n \qquad (5.13)$$

Substituting equation (5.13) in equation (5.9) and bringing all the unknowns on the left hand side one gets

$$[\tilde{I}_{n,2n} \;\; -\tilde{H}(K)] \begin{bmatrix} \tilde{\underline{x}}(K+1) \\ \tilde{\underline{u}}(K) \end{bmatrix} = \underline{P}(K) \qquad (5.14)$$

---

1) The optimization problem where the performance index is defined over one sampling interval starting from the present instant, is called a one-interval control problem.

where

$$\tilde{I}_{n,2n} = \begin{bmatrix} 1 & -1 & & & & & \\ & & 1 & -1 & & 0 & \\ & & & & \ddots & & \\ & & 0 & & & \ddots & \\ & & & & & 1 & -1 \end{bmatrix} \quad (n \times 2n) \text{ matrix}$$

$$\tilde{H}(K) = \begin{bmatrix} h_1(k) & -h_1(K) \\ h_2(K) & -h_2(K) \\ \vdots & \vdots \\ h_n(K) & -h_n(K) \end{bmatrix} \quad (n \times 2) \text{ matrix}$$

$$\underline{\tilde{x}}(K) = \begin{bmatrix} x_1^+(K+1) \\ x_1^-(K+1) \\ x_2^+(K+1) \\ x_2^-(K+1) \\ \vdots \\ x_n^+(K+1) \\ x_n^-(K+1) \end{bmatrix} \text{ 2n-vector} \quad , \quad \underline{\tilde{u}}(K) = \begin{bmatrix} u^+(K+1) \\ u^-(K+1) \end{bmatrix} \text{ 2-vector}$$

$$\underline{P}(K) = \begin{bmatrix} p_1(K) \\ p_2(K) \\ \vdots \\ p_n(K) \end{bmatrix} \text{ n-vector, } p_i(K) = \sum_{j=1}^{n} a_{ij}(K)\, x_j(K) + d_i(K) \quad . \; i = 1, 2, \ldots, n. \tag{5.15}$$

Similarly by an analogous substitution, and introduction of slack variables s_j, $j = 1, 2, \ldots, r$, as defined earlier, equation (5.10) can be replaced by

$$[\tilde{B}_{1n} \quad \tilde{B}_{n,n+1} \quad I_{r,r}] \begin{bmatrix} \tilde{\underline{x}} \\ \tilde{\underline{u}} \\ \underline{s} \end{bmatrix} = \underline{\alpha} \tag{5.16}$$

where

$$\tilde{B}_{1n} = \begin{bmatrix} b_{11} & -b_{11} & b_{12} & -b_{12} & \cdots & b_{1n} & -b_{1n} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ b_{r1} & -b_{r1} & b_{r2} & -b_{r2} & \cdots & b_{rn} & -b_{rn} \end{bmatrix} \quad (r \times 2n) \text{ matrix}$$

$$\tilde{B}_{n,n+1} = \begin{bmatrix} b_{1,n+1} & -b_{1,n+1} \\ \vdots & \vdots \\ b_{r,n+1} & -b_{r,n+1} \end{bmatrix} \quad (r \times 2) \text{ matrix}$$

$I_{r,r}$ = (r x r) identity matrix

$$\underline{s} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_r \end{bmatrix} \quad r\text{-vector} \tag{5.17}$$

Augmenting equations (5.14) and (5.16) together one gets

$$\begin{bmatrix} \tilde{I} & -\tilde{H} & 0 \\ \tilde{B}_{1n} & B_{n,n+1} & I \end{bmatrix} \begin{bmatrix} \tilde{\underline{x}} \\ \tilde{\underline{u}} \\ \underline{s} \end{bmatrix} = \begin{bmatrix} \underline{P} \\ \underline{\alpha} \end{bmatrix} \tag{5.18}$$

Let

$$A_{11} = \begin{bmatrix} \tilde{I} & -\tilde{H} & 0 \\ \tilde{B}_{1n} & B_{n,n+1} & I \end{bmatrix} \qquad \underline{y} = \begin{bmatrix} \tilde{\underline{x}} \\ \tilde{\underline{u}} \\ \underline{s} \end{bmatrix}, \quad \underline{q} = \begin{bmatrix} \underline{P} \\ \underline{\alpha} \end{bmatrix} \tag{5.19}$$

then

$$A_{11}\, \underline{y} = \underline{q}$$

$$y_j \geq 0 \quad , \quad j = 1, 2, \ldots, (2n + 2+r) \tag{5.20}$$

Note that equation (5.20) is in the standard form. Now define a new performance index to be minimized

$$\text{P.I.} = \underline{g}'\, \underline{y} \tag{5.21}$$

where

$$\underline{g}' = [\underline{\tilde{c}}' \quad \tilde{\gamma}' \quad 0] \qquad (2n + 2+r)\text{-vector}$$

$$\underline{\tilde{c}}' = [c_1 \quad c_1 \quad c_2 \quad c_2 \quad \cdots \quad c_n \quad c_n]$$

$$\tilde{\gamma}' = [\gamma \quad \gamma] \tag{5.22}$$

Equations (5.20) and (5.21) constitute a legitimate linear programming problem in the standard form. There are (2n + 2+r) unknowns and (n+r) equations.

In the coefficient matrix of equation (5.20) column 2j is negative of column (2j-1) for j = 1, 2, ..., n+1, that is the two columns are linearly dependent.. Therefore y_{2j-1} and y_{2j} for j = 1, 2, ..., n+1, can not both be basic variables; hence in the minimal feasible solution at least one of them is zero. Therefore

$$y_{2j-1} + y_{2j} = |\, y_{2j-1} + y_{2j}\,|$$

$$= |\, y_{2j-1} - y_{2j}\,|$$

or $$y_{2j-1} + y_{2j} = |\, x_j^+(K+1) - x_j^-(K+1)\,| \quad \text{for} \quad j = 1, 2, \ldots, n$$

and $$y_{2n+1} + y_{2n+2} = |\, u^+(K) - u^-(K)\,| \tag{5.23}$$

which means minimizing equation (5.21) is the same as minimizing the right hand side of equation (5.12).

Problem IV Two-Interval Control Problem

Replace the performance index in Problem III by

$$c'(K+1)\,|\,\underline{x}(K+1)\,|\,+\,\gamma(K)\,|\,u(K)\,|\,+\,c'(K+2)\,|\,\underline{x}(K+2)\,|\,+\,\gamma(K+1)\,|\,u(K+1)\,|\qquad(5.24)$$

The solution to this problem would be a <u>two-interval control</u> because the solution involves the control signal u(K) and u(K+1). The variables involved in the solution are $\underline{x}(K+1)$, $\underline{x}(K+2)$, u(K), u(K+1), $\underline{s}(K+1)$ and $\underline{s}(K+2)$ where $\underline{s}$ stands for the slack variables.

Define

$$\tilde{A} = \begin{bmatrix} a_{11} & -a_{11} & \cdots & a_{1n} & -a_{1n} \\ a_{21} & -a_{21} & \cdots & a_{2n} & -a_{2n} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & -a_{n1} & \cdots & a_{nn} & -a_{nn} \end{bmatrix} \qquad (5.25)$$

Using the notation in equations (5.15), (5.17) and (5.25) the plant equation for two intervals can be written as

$$\begin{bmatrix} \tilde{I} & 0 & -\tilde{H}(K) & 0 \\ -\tilde{A}(K+1) & \tilde{I} & 0 & -\tilde{H}(K+1) \end{bmatrix} \begin{bmatrix} \tilde{\underline{x}}(K+1) \\ \tilde{\underline{x}}(K+2) \\ \tilde{\underline{u}}(K) \\ \tilde{\underline{u}}(K+1) \end{bmatrix} = \begin{bmatrix} \underline{P}(K) \\ \underline{d}(K+1) \end{bmatrix} \qquad (5.26)$$

Similarly the equations corresponding to the constraints for two intervals would be

$$\begin{bmatrix} \tilde{B}_{1n}(K+1) & 0 & \tilde{B}_{n,n+1}(K+1) & 0 & I & 0 \\ 0 & \tilde{B}_{1n}(K+2) & 0 & B_{n,n+1}(K+2) & 0 & I \end{bmatrix}$$

(Matrix Equation Continued)

$$\begin{bmatrix} \tilde{\underline{x}}(K+1) \\ \tilde{\underline{x}}(K+2) \\ \tilde{\underline{u}}(K) \\ \tilde{\underline{u}}(K+1) \\ \underline{s}(K+1) \\ \underline{s}(K+2) \end{bmatrix} = \begin{bmatrix} \underline{\alpha}(K+1) \\ \underline{\alpha}(K+2) \end{bmatrix} \tag{5.27}$$

Augmenting equation (5.26) and equation(5.27) together

$$\begin{bmatrix} \tilde{I} & 0 & -\tilde{H}(K) & 0 & 0 & 0 \\ -\tilde{A}(K+1) & \tilde{I} & 0 & -\tilde{H}(k+1) & 0 & 0 \\ \tilde{B}_{1n}(K+1) & 0 & \tilde{B}_{n,n+1}(K+1) & 0 & I & 0 \\ 0 & \tilde{B}_{1n}(K+2) & 0 & \tilde{B}_{n,n+1}(K+2) & 0 & I \end{bmatrix}$$

$$\begin{bmatrix} \tilde{\underline{x}}(K+1) \\ \tilde{\underline{x}}(K+2) \\ \tilde{\underline{u}}(K) \\ \tilde{\underline{u}}(K+1) \\ \underline{s}(K+1) \\ \underline{s}(K+2) \end{bmatrix} = \begin{bmatrix} \underline{P}(K) \\ \underline{d}(K+1) \\ \underline{\alpha}(K+1) \\ \underline{\alpha}(K+2) \end{bmatrix} \tag{5.28}$$

Using the information in equation (5.23) and the notation of equation (5.22), the performance index in equation (5.24) can be written as

$$[\tilde{\underline{c}}'(K+1) \quad \tilde{\underline{c}}'(K+2) \quad \tilde{\underline{\gamma}}'(K) \quad \tilde{\underline{\gamma}}'(K+1) \quad 0 \quad 0] \begin{bmatrix} \tilde{\underline{x}}(K+1) \\ \tilde{\underline{x}}(K+2) \\ \tilde{\underline{u}}(K) \\ \tilde{\underline{u}}(K+1) \\ \underline{s}(K+1) \\ \underline{s}(K+2) \end{bmatrix} \tag{5.29}$$

Equations (5.28) and (5.29) together constitute a L.P. problem in the standard form for two-interval control.

Based on the development of the formulation of one- and two-interval control problems in the standard form of linear programming, generalization to N-interval control problem is straight forward.

Problem V N-Interval Control Problem

Minimize the right hand side of equation (5.11) subject to (5.9) and (5.10).

Proceeding in a manner similar to that in the case of problem III and IV, the linear programming formulation of this problem can be obtained. Using the earlier notation this is given by equations (5.30) and (5.31) on the next two pages.

Problem VI Min-Max Problem

It is possible to minimize the maximum magnitude of a linear function of state and control variables.

Consider the following optimization problem:

$$\text{Minimize} \quad \max_{K=0,1,\ldots,N-1} \left| \underline{c}'(K+1)\, \underline{x}(K+1) + \gamma(K)\, u(K) \right| \tag{5.32}$$

subject to equations (5.9) and (5.10).

$$\text{Let} \quad \max_{K=0,1,\ldots,N-1} \left| \underline{c}'(K+1)\, \underline{x}(K+1) + \gamma(K)\, u(K) \right| = Z \geq 0 \tag{5.33}$$

then

$$\left| \underline{c}'(K+1)\, \underline{x}(K+1) + \gamma(K)\, u(K) \right| \leq Z$$

or

$$\underline{c}'(K+1)\, \underline{x}(K+1) + \gamma(K)\, u(K) \leq Z \tag{5.34a}$$

$$-\underline{c}'(K+1)\, \underline{x}(K+1) - \gamma(K)\, u(K) \leq Z \tag{5.34b}$$

$$K = 0, 1, \ldots, N-1$$

Minimize

$$\begin{bmatrix} \tilde{\underline{c}}'(1) & \tilde{\underline{c}}'(2) & \cdots & \tilde{\underline{c}}'(N-1) & \tilde{\underline{c}}'(N) & \tilde{\underline{\chi}}'(0) & \tilde{\underline{\chi}}'(1) & \cdots & \tilde{\underline{\chi}}'(N-1) & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \tilde{\underline{x}}(1) \\ \tilde{\underline{x}}(2) \\ \vdots \\ \tilde{\underline{x}}(N-1) \\ \tilde{\underline{x}}(N) \\ \tilde{\underline{u}}(0) \\ \tilde{\underline{u}}(1) \\ \vdots \\ \tilde{\underline{u}}(N-1) \\ \underline{s}(1) \\ s(2) \\ \vdots \\ \underline{s}(N) \end{bmatrix} \quad (5.30)$$

Subject to

$$\begin{bmatrix} \tilde{I} & 0 & \cdots & 0 & 0 & -\tilde{H}(0) & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ -\tilde{A}(1) & \tilde{I} & \cdots & 0 & 0 & 0 & -\tilde{H}(1) & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & -\tilde{A}(2) & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & -\tilde{A}(N-1) & \tilde{I} & 0 & 0 & \cdots & -H(N-1) & 0 & 0 & \cdots & 0 \\ \tilde{B}_{1,n}(1) & 0 & \cdots & 0 & 0 & \tilde{B}_{n,n+1}(1) & 0 & \cdots & 0 & I & 0 & \cdots & 0 \\ 0 & \tilde{B}_{1,n}(2) & \cdots & 0 & 0 & 0 & \tilde{B}_{n,n+1}(2) & \cdots & 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & B_{1,n}(N) & 0 & 0 & \cdots & \tilde{B}_{n,n+1}(N) & 0 & 0 & \cdots & I \end{bmatrix} \begin{bmatrix} \tilde{x}(1) \\ \tilde{x}(2) \\ \vdots \\ \tilde{x}(N-1) \\ \tilde{x}(N) \\ \tilde{u}(0) \\ \tilde{u}(1) \\ \vdots \\ \tilde{u}(N-1) \\ s(1) \\ s(2) \\ \vdots \\ s(N) \end{bmatrix} = \begin{bmatrix} P(0) \\ d(1) \\ d(2) \\ \vdots \\ d(N-1) \\ \alpha(1) \\ \alpha(2) \\ \vdots \\ \alpha(N) \end{bmatrix} \quad (5.31)$$

$$
\begin{bmatrix}
\tilde{\underline{c}}'(1) & 0 & \cdots & 0 & \tilde{\chi}'(0) & 0 & \cdots & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 \\
-\tilde{\underline{c}}'(1) & 0 & \cdots & 0 & -\tilde{\chi}'(0) & 0 & \cdots & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & -1 \\
0 & \tilde{\underline{c}}'(2) & \cdots & 0 & 0 & \tilde{\chi}'(1) & \cdots & 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 & -1 \\
0 & -\tilde{\underline{c}}'(2) & \cdots & 0 & 0 & -\tilde{\chi}'(1) & \cdots & 0 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 & -1 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & \tilde{\underline{c}}'(N) & 0 & 0 & \cdots & \tilde{\chi}'(N-1) & 0 & 0 & 0 & 0 & \cdots & 1 & 0 & -1 \\
0 & 0 & \cdots & -\tilde{\underline{c}}'(N) & 0 & 0 & \cdots & -\tilde{\chi}'(N-1) & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -1
\end{bmatrix}
\begin{bmatrix}
\tilde{\underline{x}}(1) \\ \tilde{\underline{x}}(2) \\ \vdots \\ \tilde{\underline{x}}(N) \\ \tilde{\underline{u}}(0) \\ \tilde{\underline{u}}(1) \\ \vdots \\ \tilde{\underline{u}}(N-1) \\ \omega(1) \\ \omega(2) \\ \omega(3) \\ \omega(4) \\ \vdots \\ \omega(N-1) \\ \omega(N) \\ Z
\end{bmatrix}
= 0 \qquad (5.35)
$$

Using the notation defined earlier and introducing the slack variables ω, equations (5.34a) and (5.34b) together can be replaced by equation (5.35) for all K.

Let

$$A^* = \begin{bmatrix} \tilde{I} & 0 & \cdots & 0 & 0 \\ -\tilde{A}(1) & \tilde{I} & \cdots & 0 & 0 \\ 0 & -\tilde{A}(2) & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & -\tilde{A}(N-1) & \tilde{I} \end{bmatrix} \qquad (5.36a)$$

$$B^* = \begin{bmatrix} \tilde{B}_{1,n}(1) & 0 & \cdots & 0 & 0 \\ 0 & \tilde{B}_{1,n}(2) & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \tilde{B}_{1,n}(N) \end{bmatrix} \qquad (5.36b)$$

$$H^* = \begin{bmatrix} -\tilde{H}(0) & 0 & \cdots & 0 \\ 0 & -\tilde{H}(1) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & -\tilde{H}(N-1) \end{bmatrix} \qquad (5.36c)$$

$$G^* = \begin{bmatrix} \tilde{B}_{n,n+1}(1) & 0 & \cdots & 0 \\ 0 & \tilde{B}_{n,n+1}(2) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \tilde{B}_{n,n+1}(N) \end{bmatrix} \qquad (5.36d)$$

$$\underline{X}^* = \begin{bmatrix} \underline{\tilde{x}}(1) \\ \underline{\tilde{x}}(2) \\ \vdots \\ \underline{\tilde{x}}(N) \end{bmatrix}, \quad \underline{U}^* = \begin{bmatrix} \underline{\tilde{u}}(0) \\ \underline{\tilde{u}}(1) \\ \vdots \\ \underline{\tilde{u}}(N-1) \end{bmatrix}, \quad \underline{S}^* = \begin{bmatrix} \underline{s}(1) \\ \underline{s}(2) \\ \vdots \\ \underline{s}(N) \end{bmatrix}$$

$$\underline{D}^* = \begin{bmatrix} \underline{P}(0) \\ \underline{d}(1) \\ \vdots \\ \underline{d}(N-1) \end{bmatrix}, \quad \underline{\alpha}^* = \begin{bmatrix} \underline{\alpha}(1) \\ \underline{\alpha}(2) \\ \vdots \\ \underline{\alpha}(N) \end{bmatrix} \tag{5.36e}$$

Then substituting equation (5.36) in equation (5.31)

$$\begin{bmatrix} A^* & H^* & 0 \\ B^* & G^* & I \end{bmatrix} \begin{bmatrix} \underline{X}^* \\ \underline{U}^* \\ \underline{S}^* \end{bmatrix} = \begin{bmatrix} \underline{D}^* \\ \underline{\alpha}^* \end{bmatrix} \tag{5.37}$$

Now let

$$\underline{\delta}^* = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix} \quad \text{2N-vector} \tag{5.38a}$$

$$C^* = \begin{bmatrix} \underline{\tilde{c}}'(1) & 0 & \cdots & 0 \\ -\underline{\tilde{c}}'(1) & 0 & \cdots & 0 \\ 0 & \underline{\tilde{c}}'(2) & \cdots & 0 \\ 0 & -\underline{\tilde{c}}'(2) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \underline{\tilde{c}}'(N) \\ 0 & 0 & \cdots & -\underline{\tilde{c}}'(N) \end{bmatrix} \tag{5.38b}$$

$$\Gamma^* = \begin{bmatrix} \tilde{\chi}'(0) & 0 & \cdots & 0 \\ -\tilde{\chi}'(0) & 0 & \cdots & 0 \\ 0 & \tilde{\chi}'(1) & \cdots & 0 \\ 0 & -\tilde{\chi}'(1) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \tilde{\chi}'(N-1) \\ 0 & 0 & \cdots & -\tilde{\chi}'(N-1) \end{bmatrix} \tag{5.38c}$$

$$\underline{\Omega} = \begin{bmatrix} \omega(1) \\ \omega(2) \\ \vdots \\ \omega(2N) \end{bmatrix} \tag{5.38d}$$

then equation (5.35) can be written as

$$[C^* \quad \Gamma^* \; I \quad \underline{\delta}^*] \begin{bmatrix} \underline{X}^* \\ \underline{U}^* \\ \underline{\Omega}^* \\ Z \end{bmatrix} = 0 \tag{5.39}$$

Augmenting equation (5.39) to equation (5.37) in a suitable manner one gets

$$\begin{bmatrix} A^* & H^* & 0 & 0 & 0 \\ B^* & G^* & I & 0 & 0 \\ C^* & \Gamma^* & 0 & I & \underline{\delta}^* \end{bmatrix} \begin{bmatrix} \underline{X}^* \\ \underline{U}^* \\ \underline{S}^* \\ \underline{\Omega}^* \\ Z \end{bmatrix} = \begin{bmatrix} \underline{D}^* \\ \underline{\alpha}^* \\ 0 \end{bmatrix} \tag{5.40}$$

Now equation (5.40) incorporates equations (5.9), (5.10) and (5.34) for all K. Thus the minimization of Z subject to equation (5.40) which is in the standard form of linear programming, is the solution to Min-Max problem.

5.5 Modified Optimum Control Policy

If T is the time interval over which the performance of a system is of interest, then it seems natural to perform optimization over $N = \frac{T}{\Delta}$ sampling intervals (Δ is the sampling period).

A cursory look at problem V indicates that for an N-stage process the number of variables and the equations involved in the L.P. formulation is N(2n+r+1) and N(n+r) respectively, where n is the order of the system and r is the number of inequality constraints. In the case of many realistic problems (Model Vehicle No. 2 included) these numbers are so large that for solution of these problems the number of rapid access memory locations available for storage and arithmetical and logical operations in existing computers (IBM 7094, for example) is inadequate. Also as the size of the coefficient matrix in the L.P. problem gets large, the truncation error in the computational process becomes more significant.

Now the question is: Is it necessary to define the performance index over the entire interval of interest, irrespective of how long this interval is?

The exact answer to this question is not known. The computational difficulties indicated above have led to the investigation of this problem. As a consequence a new concept of optimum control policy has developed. Under this control policy, which shall be referred to as modified optimum control policy, a control signal at the present sampling instant is

determined based on the behavior of the system at the present and a number of future sampling instants. For a linear plant and absolute value type performance index, the optimum control problem based on a modified optimum control policy can be stated as follows:

Problem VII Problem based on Modified Optimal Control Policy

For a system described by equations (5.9) and (5.10) determine u(K) such that

$$\sum_{K=K_1}^{K=(M+K_1-1)} [\underline{c}'(K+1) \mid \underline{x}(K+1) \mid + \gamma(K) \mid u(K) \mid] \qquad (5.41)$$

$$K_1 = 0, 1, \ldots, N-1$$

is minimum. $\underline{c}$, γ and N are as defined before. (M-1) is the number of future sampling instants taken under consideration to determine the control signal for the present instant.

Here an absolute-value type performance index is chosen because the associated problem is amenable to linear programming. Otherwise any convex performance index is in order.

The reasons in favor of the modified optimum control policy are threefold: i) The optimization problem is computationally feasible. ii) For linear systems with quadratic performance index, a control law can be found which is a linear function of the state variables. If the plant and the performance index are time-invariant, then the control law is also constant. This is true for both free and fixed time problems. iii) For systems with unpredictable disturbances an on-line computer control, based on the modified optimum control policy, is possible.

The computational results for Model Vehicle No. 2 presented in section 5.7 are based on the modified optimum control policy.

5.6 Treatment of Continuous Systems

So far only a discrete system has been considered. The dynamics of Model Vehicle No. 2 is described by a set of differential equations. In order to apply linear programming to Model Vehicle No. 2, it is necessary that a difference equation approximation to the differential equation be obtained.

Consider a system described by

$$\frac{d\underline{x}}{dt} = F(t)\ \underline{x}(t) + \underline{b}(t)\ u(t) + \underline{f}(t) \tag{5.42}$$

The solution to this system is given by

$$\underline{x}(t) = A(t - t_o)\ \underline{x}(0) + \int_{t_o}^{t} A(t - \tau)\ \underline{b}(\tau)\ u(\tau)\ d\tau + \int_{t_o}^{t} A(t - \tau)\ \underline{f}(\tau)\ d\tau \tag{5.43}$$

where A is the state transition matrix corresponding to the coefficient matrix F and is a solution to

$$\frac{d\ A(t)}{dt} = F(t)\ A(t)\ ,\quad A(t_o) = I \tag{5.44}$$

t_o is the initial time and $\underline{x}(t_o)$ the initial state. Now if the solution at discrete time intervals is desired and if u(t) is assumed to be piecewise constant over these intervals, then equation (5.43) can be written as

$$x(t_{K+1}) = A(t_{K+1}-t_K)\ \underline{x}(t_K) + u(t_K) \int_{t_K}^{t_{K+1}} A(t_{K+1}-\tau)\ \underline{b}(\tau)\ d\tau + \int_{t_K}^{t_{K+1}} A(t_{K+1} - \tau)\ \underline{f}(\tau)\ d\tau \tag{5.45}$$

$$K = 0,\ 1,\ 2,\ \ldots$$

If the sampling period $\Delta = (t_{K+1} - t_K)$ is small, then F, $\underline{b}$ and $\underline{f}$ can be assumed to be piecewise constant. This assumption greatly simplifies the evaluation of the right hand side of equation (5.45). Since then the state transition matrix and the integrals in equation (5.45) can be written as

$$A(t_{K+1} - t_K) \triangleq A(K) = e^{F(K)\Delta}$$
$$= I + F(K)\Delta + \frac{F^2(K)\Delta^2}{2!} + \frac{F^3(K)\Delta^3}{3!} + \ldots \qquad (5.46)$$

$$\int_{t_K}^{t_{K+1}} A(t_{K+1} - \tau)\, \underline{b}(\tau)\, d\tau \triangleq \underline{h}(K) = \underline{b}(K)\Delta + F(K)\, \underline{b}(K)\, \Delta^2$$
$$+ \frac{F^2(K)\, b(K)\Delta^3}{2!} + \ldots \qquad (5.47)$$

and

$$\int_{t_K}^{t_{K+1}} A(t_{K+1} - \tau)\, \underline{f}(\tau)\, d\tau \triangleq \underline{d}(K) = \underline{f}(K)\Delta + F(K)\, \underline{f}(K)\Delta^2$$
$$+ \frac{F^2(K)\, f(K)\Delta^3}{2!} + \ldots \qquad (5.48)$$

For small values of Δ these series converge very rapidly. In all the examples considered in the next section, for $\Delta = 0.1$ the number of terms required for the remainder to be less than 10^{-7} is below ten. Substituting equations (5.46), (5.47) and (5.48) in equation (5.45) one gets

$$\underline{x}(K+1) = A(K)\, \underline{x}(K) + \underline{h}(K)\, u(K) + \underline{d}(K) \qquad (5.49)$$

Above, for notational convenience t_K and t_{K+1} are replaced by K and (K+1) respectively. Notice the similarity between equations (5.9) and (5.49).

5.7 Computational Results

The purpose of the computational results presented here is threefold: 1) to elucidate the concept of modified optimal control policy; 2) to demonstrate how this policy can be applied to the control of Model Vehicle No. 2; 3) to relate the computational experience gained.

Example 5.1: To start with, a simple second order system is considered. This example is primarily meant to bring forth the salient features of the modified optimal control policy (considered over two intervals) as compared to the ordinary policy (one- and two-interval control) as spelled out in problem III and IV.

The plant is described by

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -x_1 - x_2 + u \tag{5.50}$$

The initial conditions are

$$x_1(0) = 0.8$$
$$x_2(0) = 0.8 \tag{5.51}$$

It is desired to bring the state of the system to the origin satisfying the following specifications:

$$|u| \leq 1$$
$$|x_1| \leq 1 \tag{5.52}$$

A discrete version of equation (5.50) for a sampling period of 0.1 sec. is given by

$$\begin{bmatrix} x_1(K+1) \\ x_2(K+1) \end{bmatrix} = \begin{bmatrix} 0.995 & 0.95 \\ -0.095 & 0.9 \end{bmatrix} \begin{bmatrix} x_1(K) \\ x_2(K) \end{bmatrix} + \begin{bmatrix} 0.0048 \\ 0.095 \end{bmatrix} u(K) \tag{5.53}$$

For this simple problem, it is possible, by trial and error, to find a large set of inputs, u(t), which satisfy the constraint (5.52) and bring the state of the system to the origin. To be precise, as is discussed in Chapter II, it is necessary to have a criterion to judge which control is the best. In optimal control theory as it stands today, the exact functional relationship between the desired goal and the performance criterion is still an open field or research. The performance index for a particular system is chosen based on intuitive feeling and

the computational experience with the system. The choice of the control policy also has a bearing on the choice of the performance index.

In the above example the goal is to get to the origin. When x_1 and x_2 are both zero, the goal is achieved. Thus it seems reasonable to minimize a linear combination of the absolute values of x_1 and x_2.

For a comparative study the example is solved for one-interval, two-interval and two-interval modified optimum control policy. Two different performance indices are considered.

1. (a) Minimize $|x_1(K+1)|$ for one-interval control
   (b) Minimize $|x_1(K+2)|$ for two-interval control
2. (a) Minimize $|x_1(K+1)| + |x_2(K+1)|$ for one-interval control
   (b) Minimize $|x_1(K+2)| + |x_2(K+2)|$ for two-interval control

Figures 5.1 and 5.2 show x_1 , x_2 and u as functions of time for one-interval control with performance indices 1(a) and 2(a) respectively. Note that when x_2(i.e., $\dot{x}_1$) is not present in the performance index, the system response is very oscillatory. And when x_2 is included in the performance index, it slows down the rate at which the performance index is minimized.

The results of two-interval control with the performance index 1(b) are shown in Figure 5.3. There is a noticeable difference between the responses shown in Figures 5.1 and 5.3. This is because for the results in Figure 5.1 the control signal at each instant is chosen with no regard to the future behaviour of the system. On the contrary for the results in Figure 5.3 the choice of the first control signal is made based on the system behaviour over two intervals and the choice of the second control signal based on only one (second) interval.

Figure 5.4 is the result of modified optimal control policy with the performance index 1(b). There is a further improvement in the results

Legend: * : x_1, 0 : x_2, R : Common.

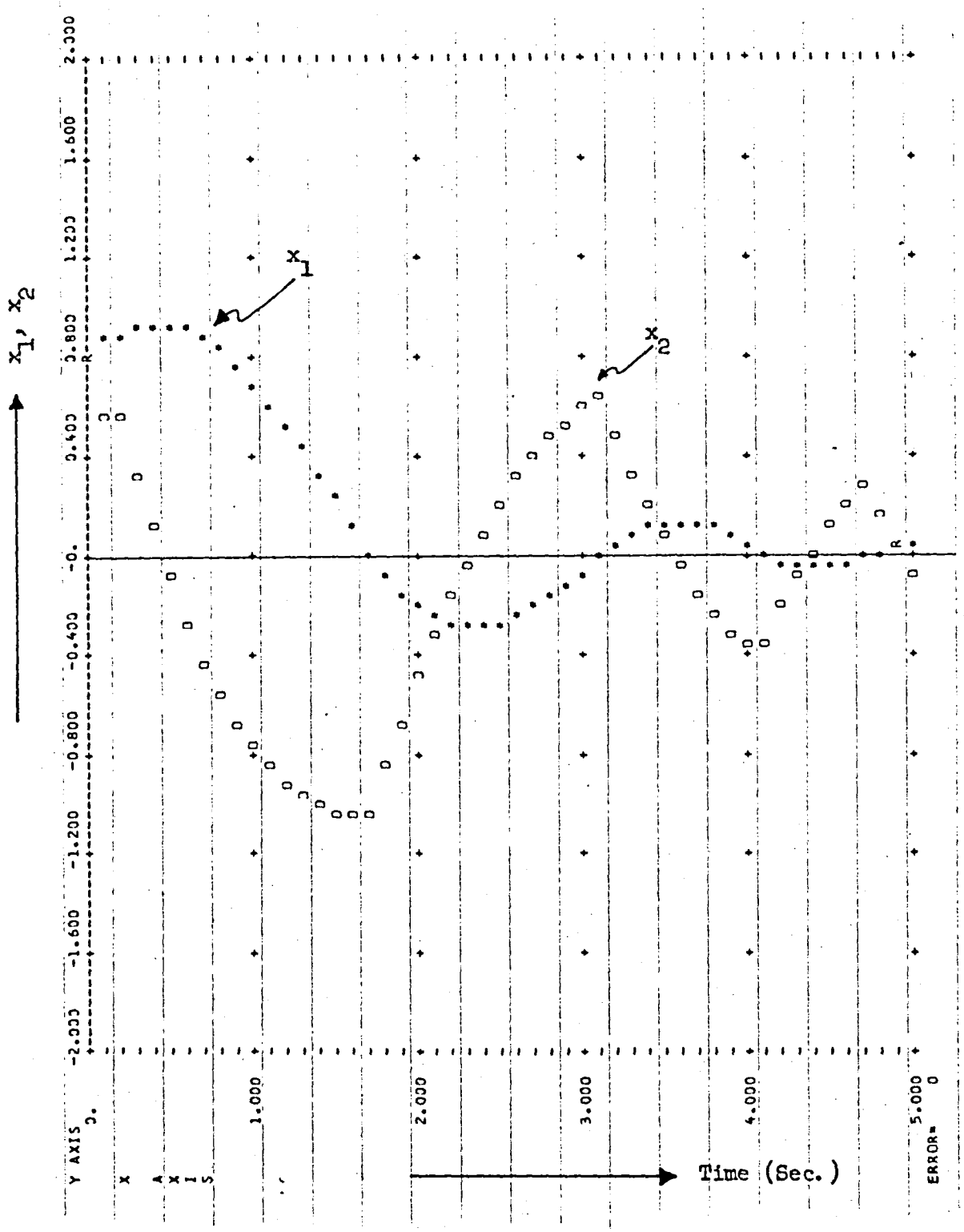


Figure 5.1 Response of the System in Example 5.1 with Performance Index (P.I.): Min $|x_1(K+1)|$ (One-interval Control).

(a) x_1 vs. time, x_2 vs. time.

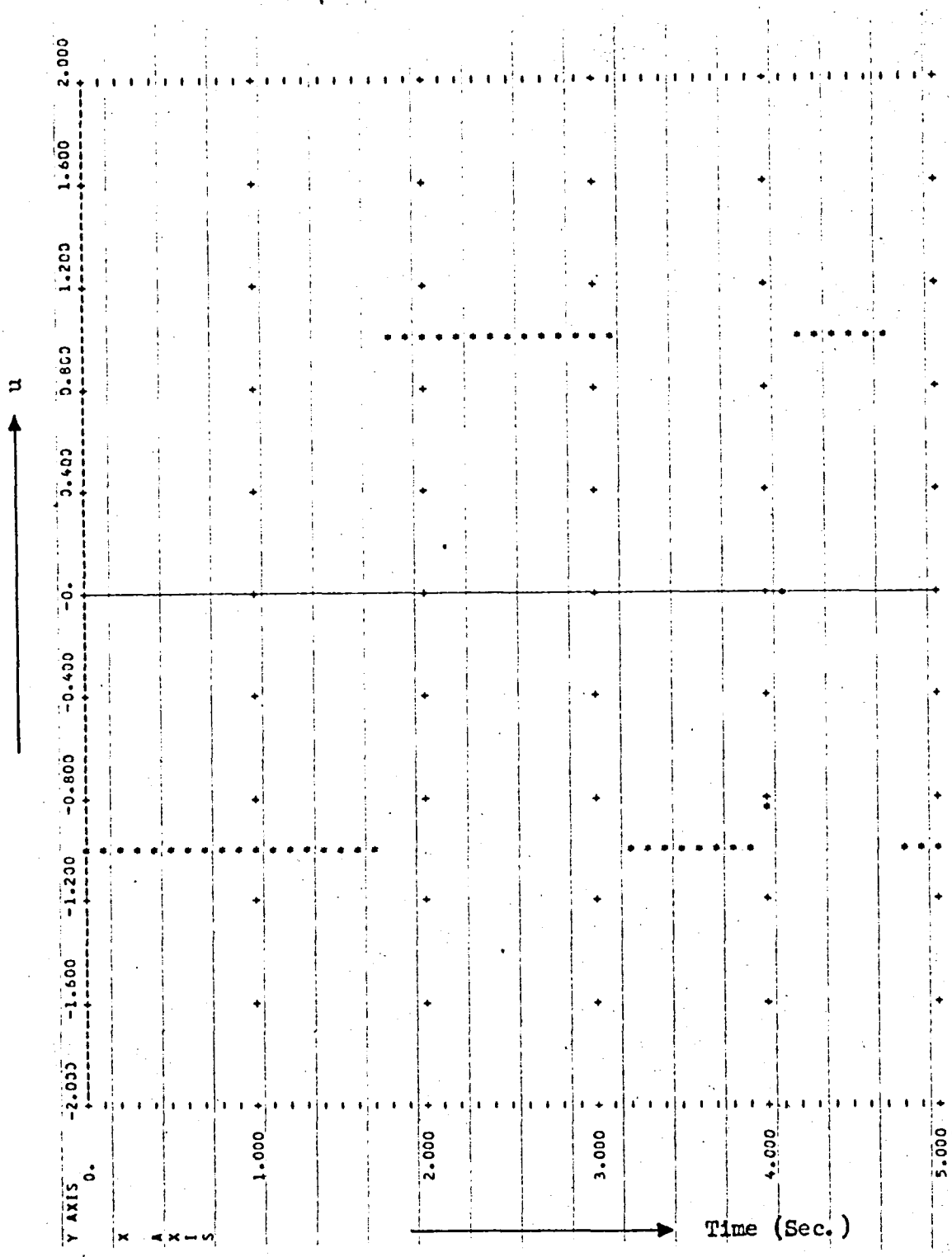


Figure 5.1 Response of the System in Example 5.1 with Performance Index (P.I.): Min $|x_1(K+1)|$ (One-interval Control).

(b) u vs. time.

Legend: * : x_1, 0 : x_2, R : Common.

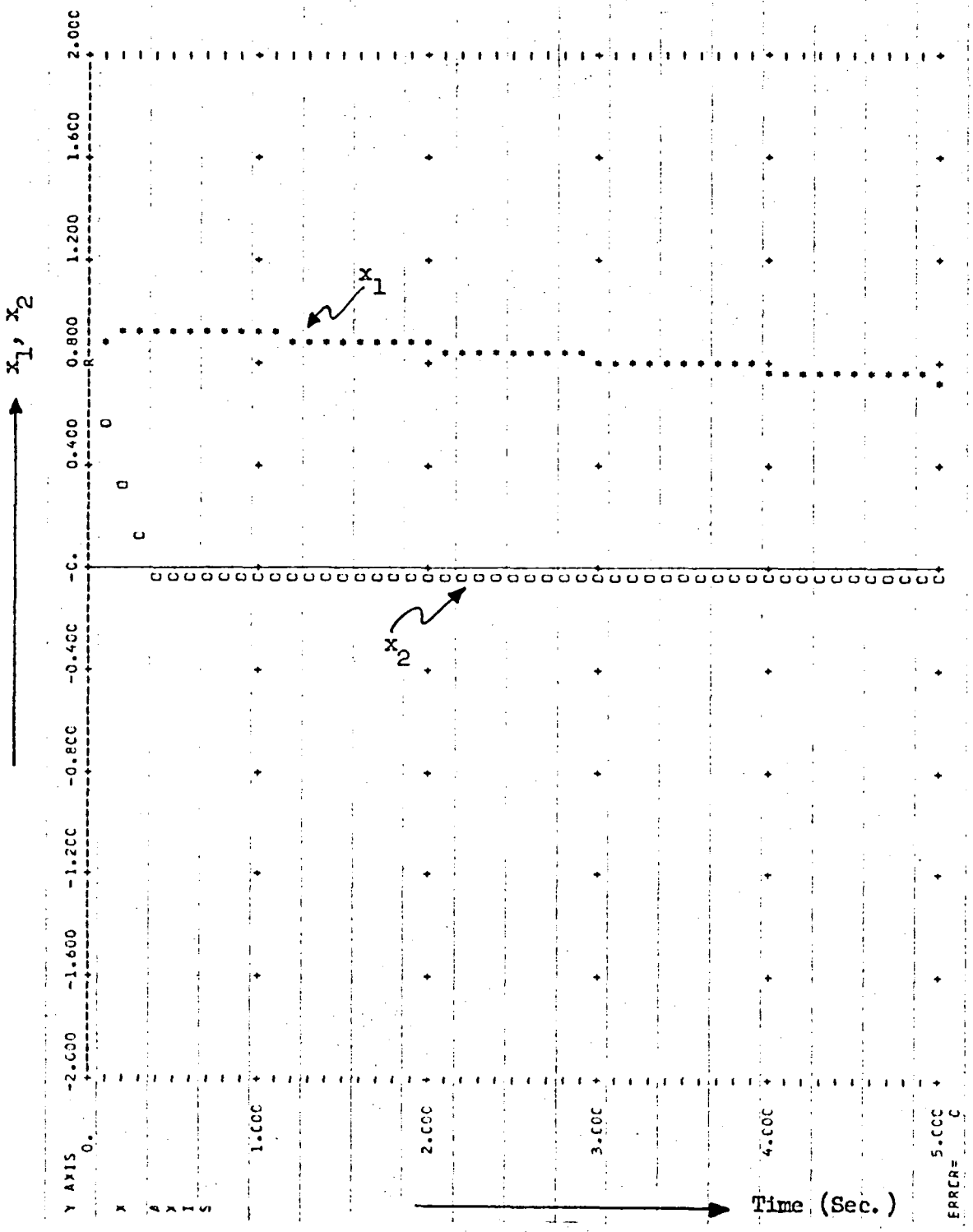


Figure 5.2 Response of the System in Example 5.1 with P.I.: Min [$|x_1(K+1)| + |x_2(K+1)|$] (One-interval Control).

(a) x_1 vs. u, x_2 vs. u .

Legend: * : u, O : P.I.

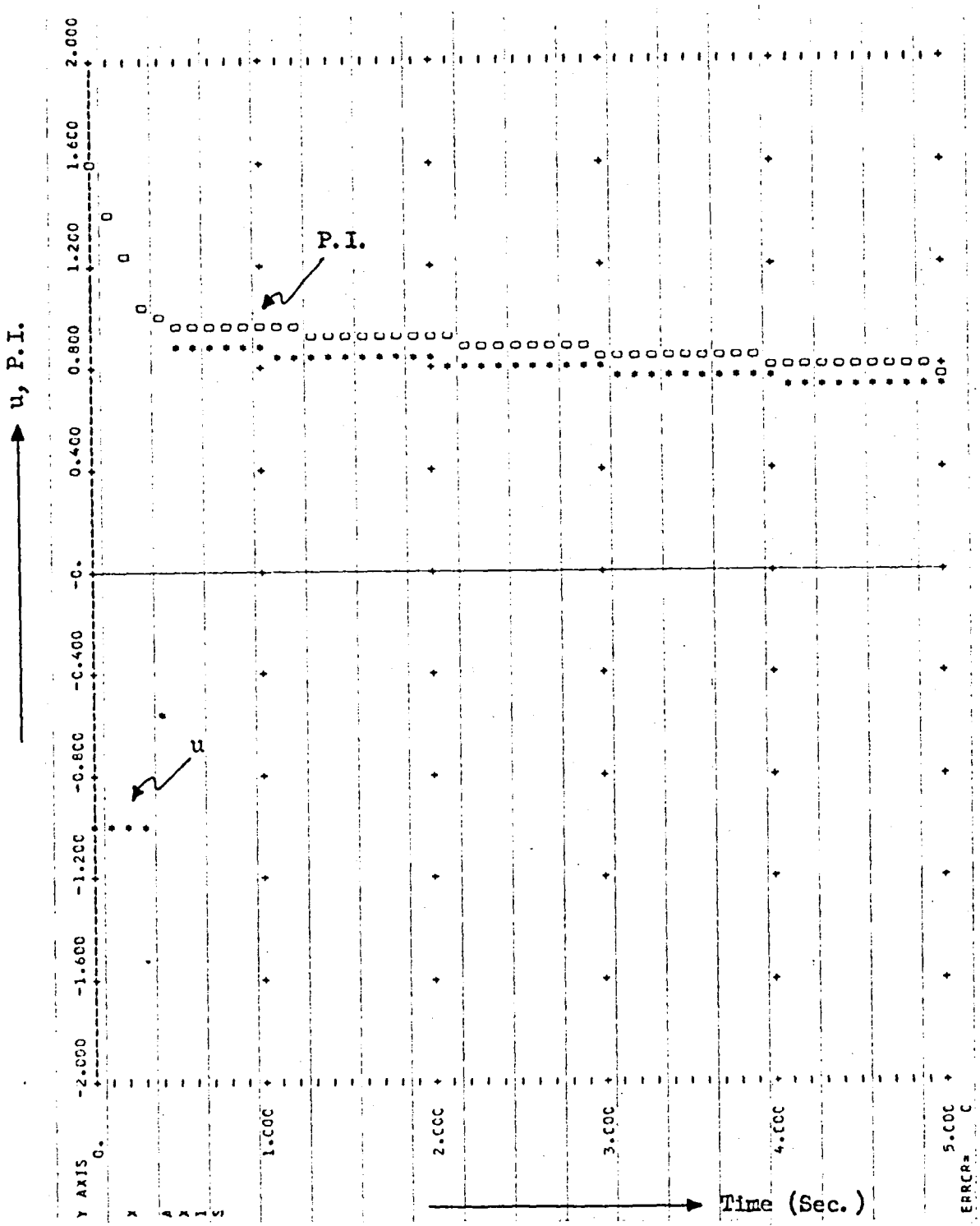


Figure 5.2 Response of the System in Example 5.1 with P.I.: Min [$|x_1(K+1)|$ + $|x_2(K+1)|$] (One-interval Control).

(b) u vs. time, P.I. vs. time.

Legend: * : x_1, 0 : x_2, R : Common.

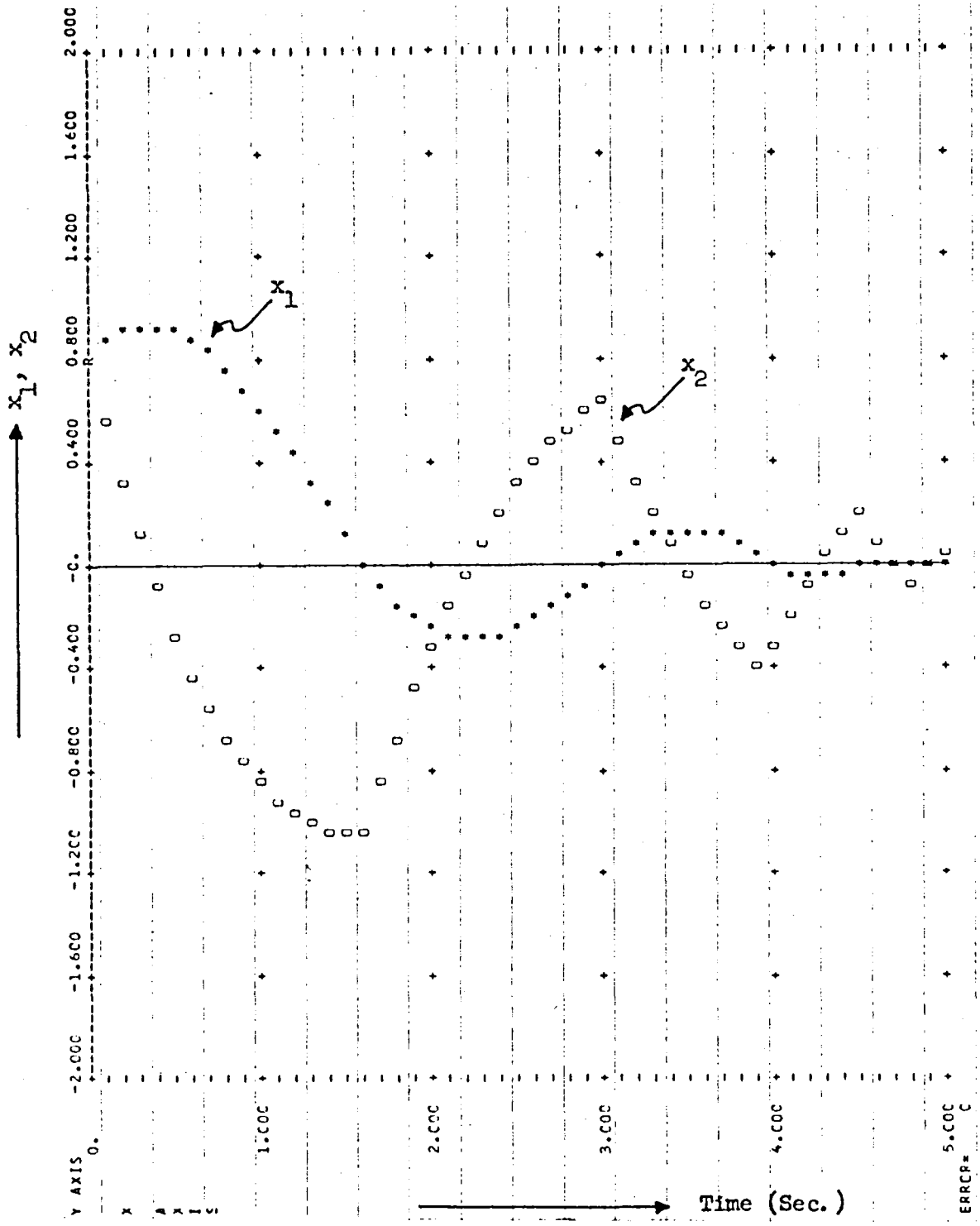


Figure 5.3 Response of the System in Example 5.1 with P.I.: Min [| $x_1(K+2)$ |] (Two-interval Control).

(a) x_1 vs. time, x_2 vs. time.

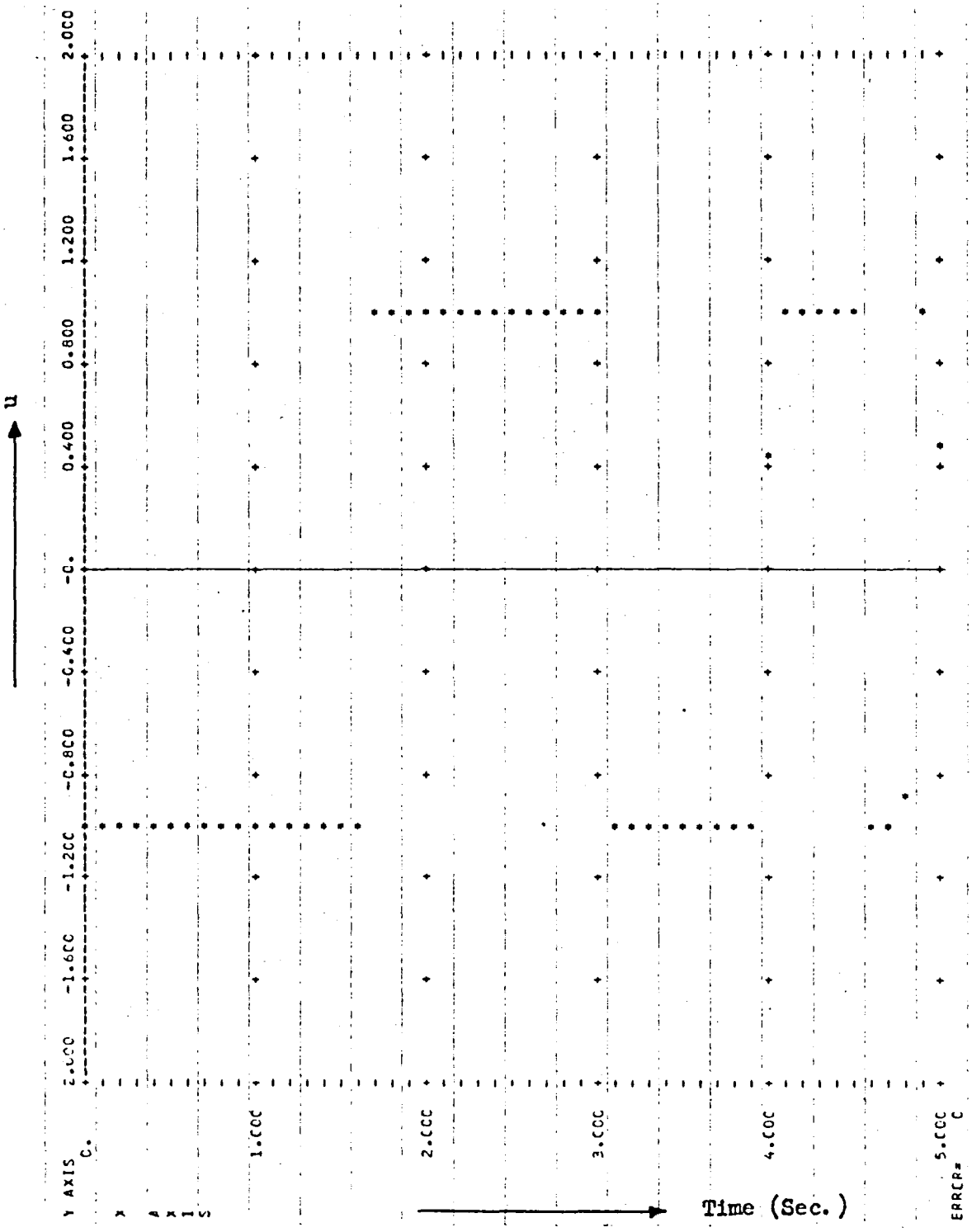


Figure 5.3 Response of the System in Example 5.1 with P.I.: Min [$|x_1(K+2)|$] (Two-interval Control).

(b) u vs. time.

Legend: * : x_1, 0 : x_2, R : Common.

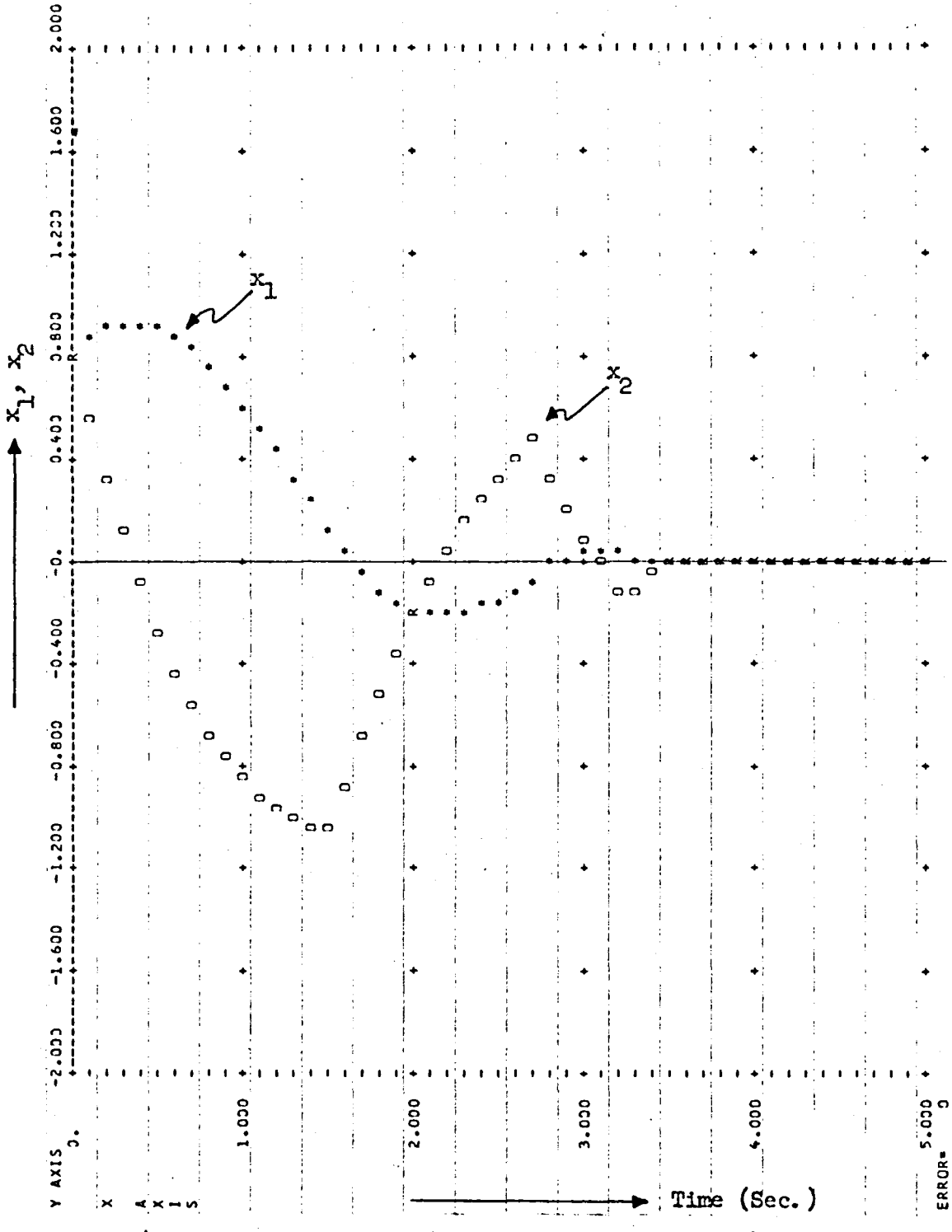


Figure 5.4 Response of the System in Example 5.1 with P.I.: Min [| $x_1(K+2)$ |] under Modified Optimal Control Policy (MOCP).

(a) x_1 vs. time, x_2 vs. time.

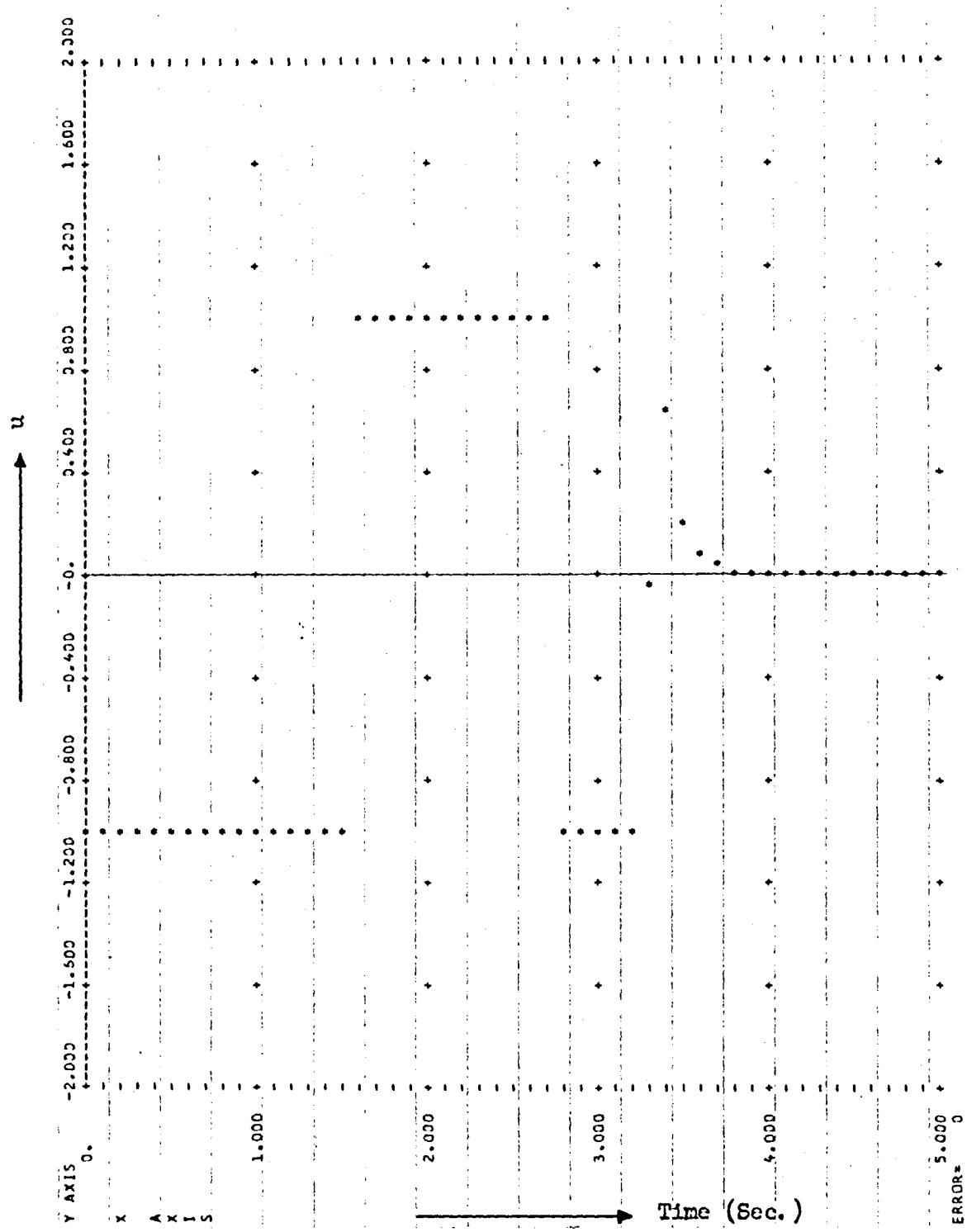


Figure 5.4 Response of the System in Example 5.1 with P.I.:
Min [| $x_1(K+2)$ |] under Modified Optimal Control Policy (MOCP)

(b) u vs. time.

in that the time taken to reach the origin is smaller and there is a smaller number of oscillations. Unlike in Figure 5.3, in Figure 5.4 the second control signal is chosen just as is the first, looking one step ahead.

Finally, Figure 5.5 shows the system response when the performance index 2(b) is minimized under modified optimal control policy. Notice again that with x_2 in the performance index the rate at which the performance index decreases becomes small.

The example above is one of several studied to investigate system performance under the modified optimal control policy. Two observations are made:

1. System performance is strongly dependent on what the quantization interval is. In the above example, if the quantization interval is reduced from 0.1 to 0.01, the system response does not remain the same. It becomes oscillatory.
2. If it is desired to bring the state of a system described in phase variable formulation to zero, all the derivatives must be weighted very lightly in the performance index. Heavy weighting of the derivatives in the performance index has a tendency to stagnate the state of the system close to the existing state, or the rate at which the performance index decreases becomes small.

Example 5.2: This is an example of an optimum control problem where the optimal solution obtained by a gradient method is known [7]. Here it is solved using linear programming under modified optimal control policy. The problem as stated by Ho and Brentani is as follows:

Legend: * : x_1, 0 : x_2, R : Common.

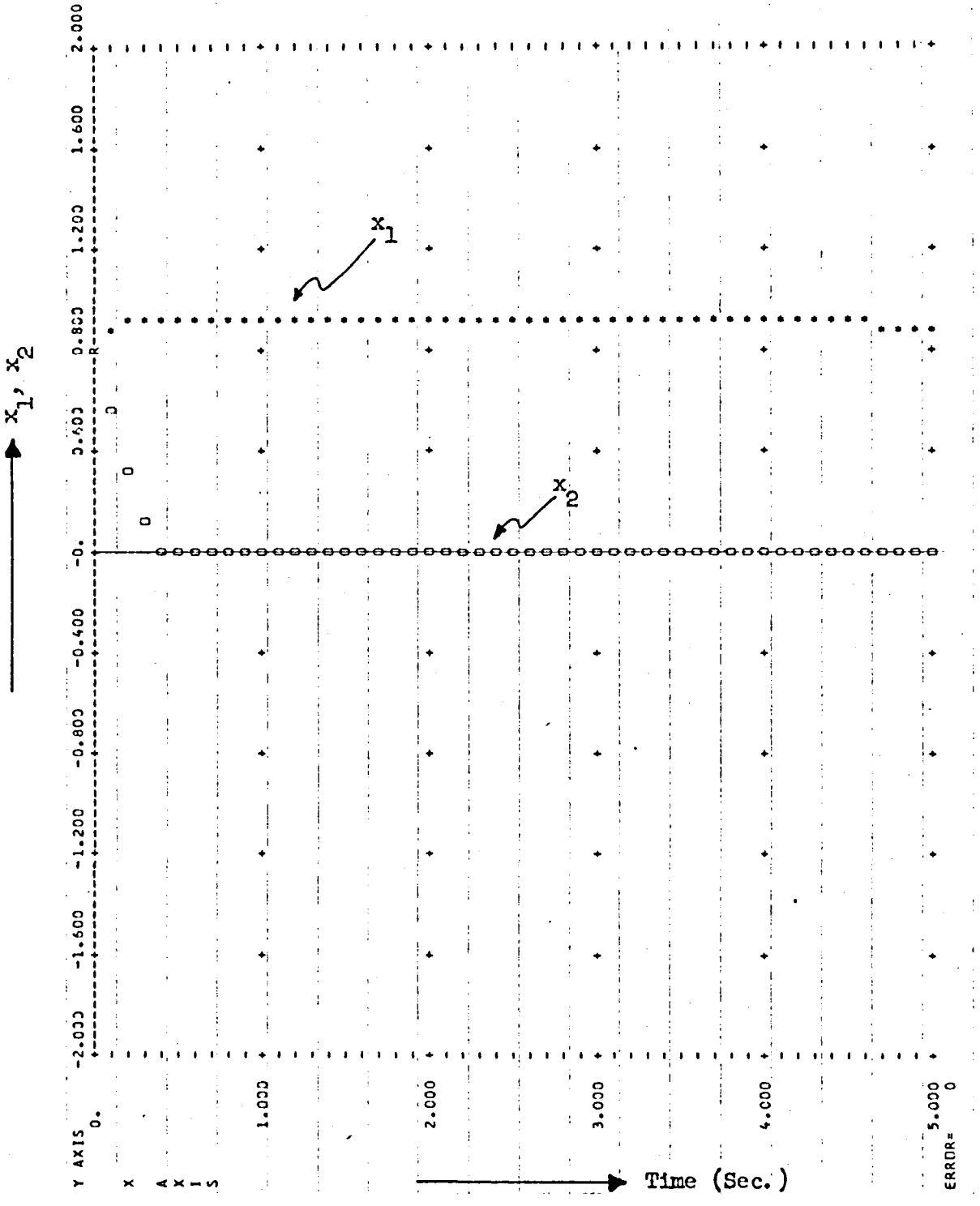


Figure 5.5 Response of the System in Example 5.1 with P.I.: Min $[\,|x_1(K+2)| + |x_2(K+2)|\,]$ under MOCP.

(a) x_1 vs. time, x_2 vs. time.

Legend: * : u, O : P.I., R : Common.

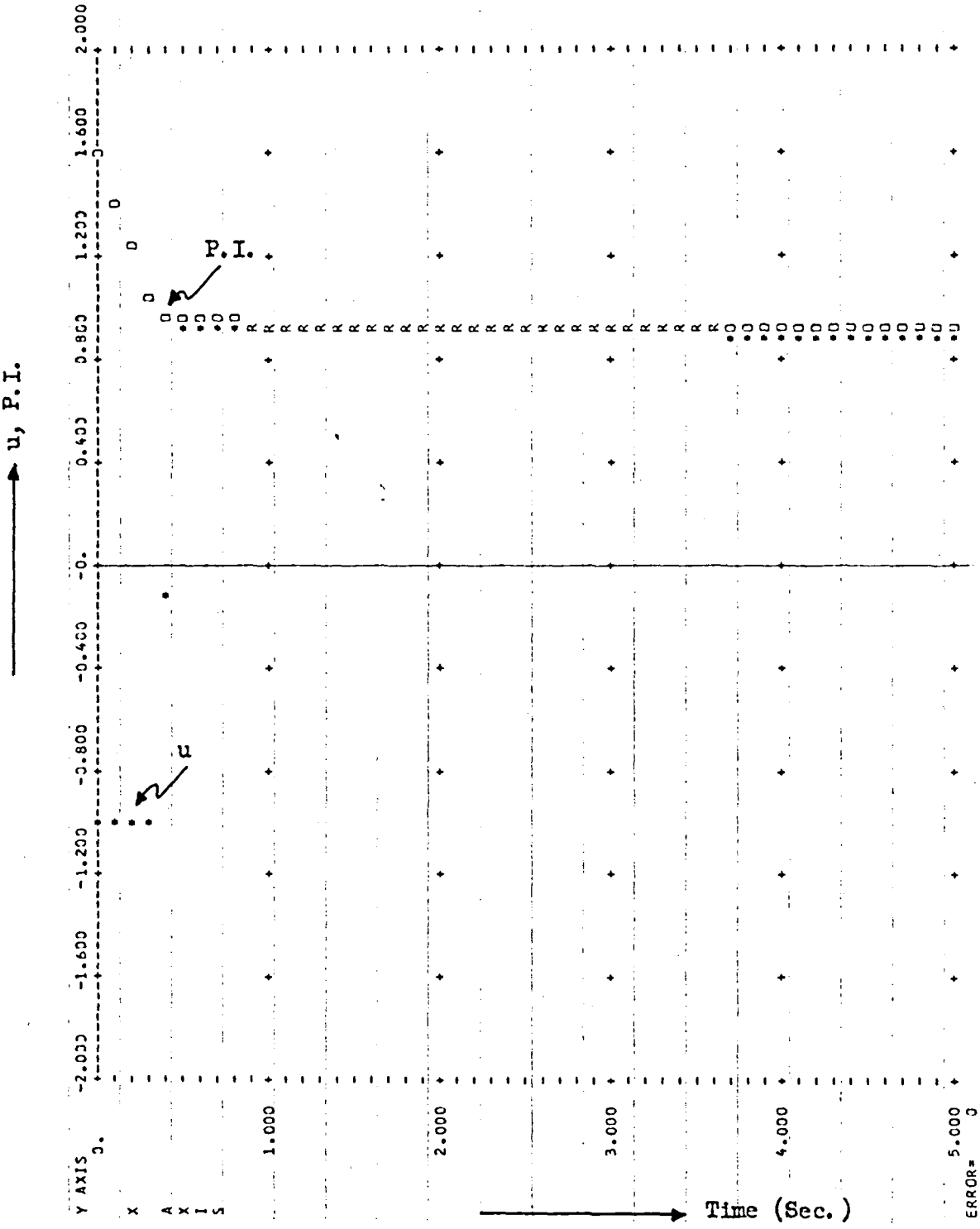


Figure 5.5 Response of the System in Example 5.1 with P.I.:
Min $[\,|x_1(K+2)| + |x_2(K+2)|\,]$ under MOCP.

(b) u vs. time, P.I. vs. time.

Given the plant

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -4 & -10 & -10 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ -4.5 \end{bmatrix} u \qquad (5.54)$$

$$\begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ x_4(0) \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (5.55)$$

and the constraints

(a) $|u| \leq 1$

(b) $|x_4| \leq 1$

find u and $\underline{x}$ such that

$$\text{P.I.}^{1)} = \text{Min} \quad [4x_1^2(t_1) + 3x_2^2(t_1) + 2x_3^2(t_1) + x_4^2(t_1)] \qquad (5.56)$$

is minimized, where $t_1 = 2.5$ seconds.

The performance index chosen by Ho and Brentani is not amenable to the linear programming approach. Keeping in mind the observations made earlier regarding the choice of the performance index under modified optimal control policy, the following performance index is chosen:

$$\text{P.I.} = \text{Min} \quad |x_1(K+2)| \qquad (5.57)$$

---

1) The performance index is not given in Ref. [7]. It was obtained by a telephone conversation with Dr. Ho.

The results obtained by Ho and Brentani and by using two-interval control under modified optimal control policy are presented in Figures 5.6 and 5.7 respectively for case (a) and Figures 5.8 and 5.9 respectively for case (b). The discretization interval used is 0.05 seconds. In spite of the fact that two sampling periods are considered at a time during the optimization process, the results are similar.

In the remainder of this section several problems on the rigid vehicle with time-invariant and time-varying coefficients with different performance indices are considered. The computational algorithms and the computer programs in Fortran for solution of these problems are given in the appendix C. The choice of the performance indices is not entirely arbitrary. Except in example 5 the goal is to stay as close to the desired trajectory as possible. In example 5 the bending moment is minimized.

The pertinent rigid body equations as given in M.S.F.C. memo on "Transfer functions for a rigid Vehicle with the General Linear Control Law" by Redus, are

$$\ddot{\phi} + c_1\alpha + c_2\beta = 0 \tag{5.58}$$

$$c_5\phi + c_3\alpha + c_4\beta - \ddot{Z} = 0 \tag{5.59}$$

$$\alpha - \phi + c_7\dot{Z} + c_4\beta = \alpha_w \tag{5.60}$$

The variables and the coefficients above are defined in the appendix B. Let

$$x_1 = \phi, \quad x_2 = \dot{\phi}, \quad x_3 = \dot{Z}, \; u = \beta, \quad v = \alpha_w \tag{5.61}$$

Then eliminating α from (5.58) and (5.59) one gets

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -c_1 & 0 & c_1c_7 \\ (c_3+c_5) & 0 & -c_3c_7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ -c_2 \\ c_4 \end{bmatrix} u + \begin{bmatrix} 0 \\ -c_1 \\ c_3 \end{bmatrix} v \tag{5.62}$$

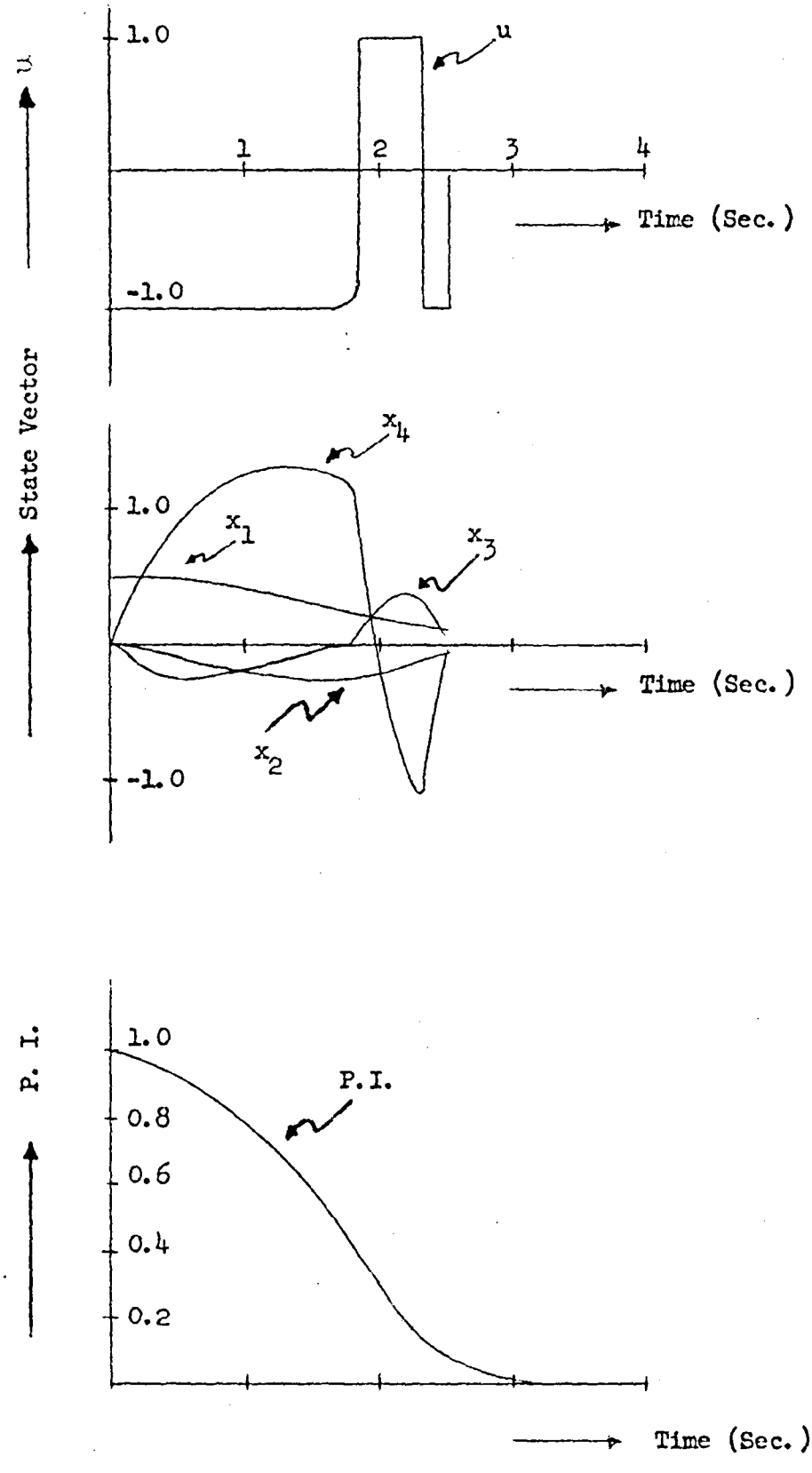


Figure 5.6

Ho and Brentani Solution to Example 5.2 with the Constraint $|u| \leq 1$

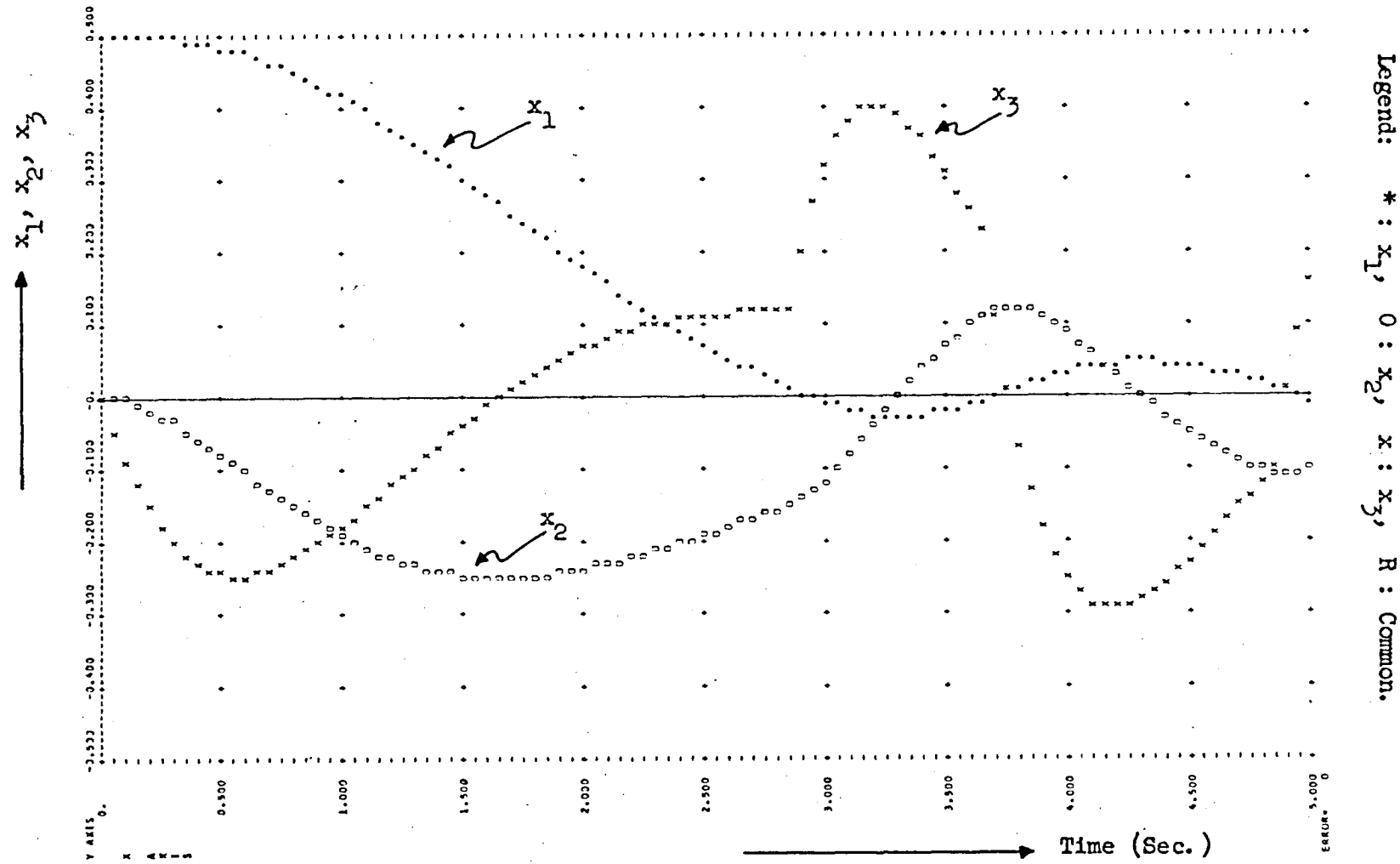


Figure 5.7 Solution to Example 5.2 under MOCP with the Constraint $|u| \leq 1$

(a) x_1 vs. time, x_2 vs. time, x_3 vs. time.

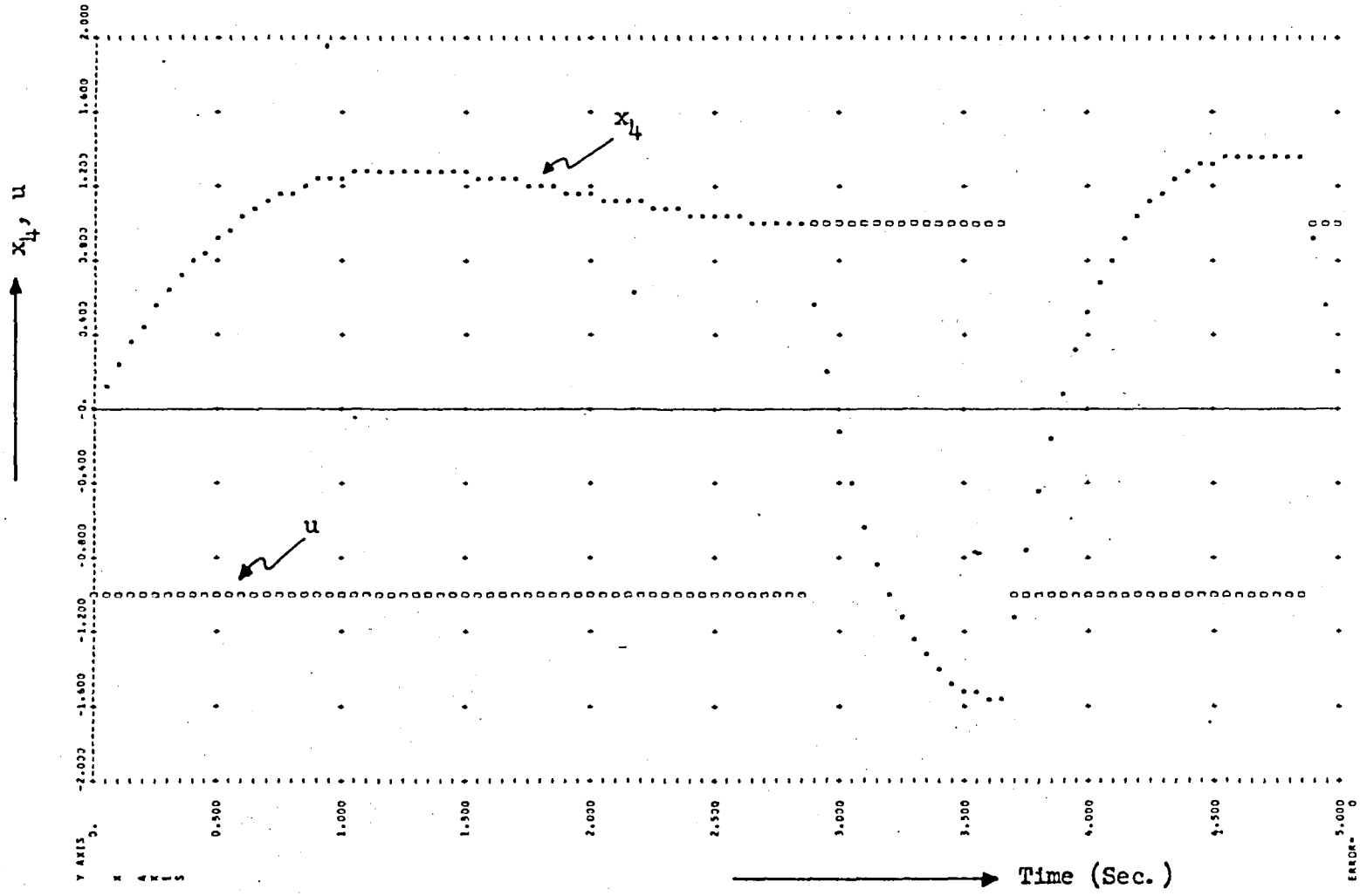


Figure 5.7 Solution to Example 5.2 under MOCP with Constraint $|u| \leq 1$

(b) x_4 vs. time, u vs. time.

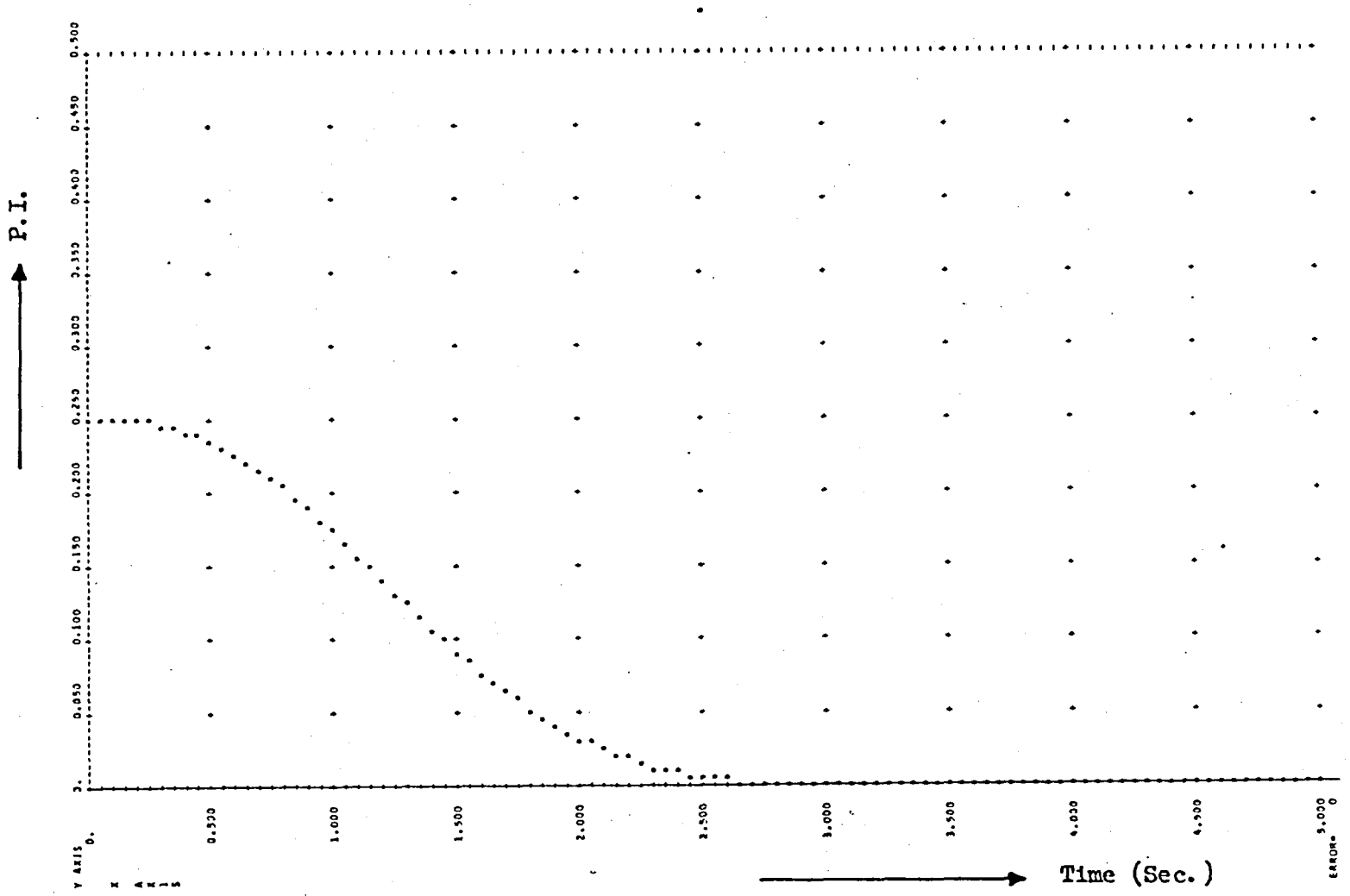


Figure 5.7 Solution to Example 5.2 under MOCP with Constraint $|u| \leq 1$

(c) P.I. vs. time.

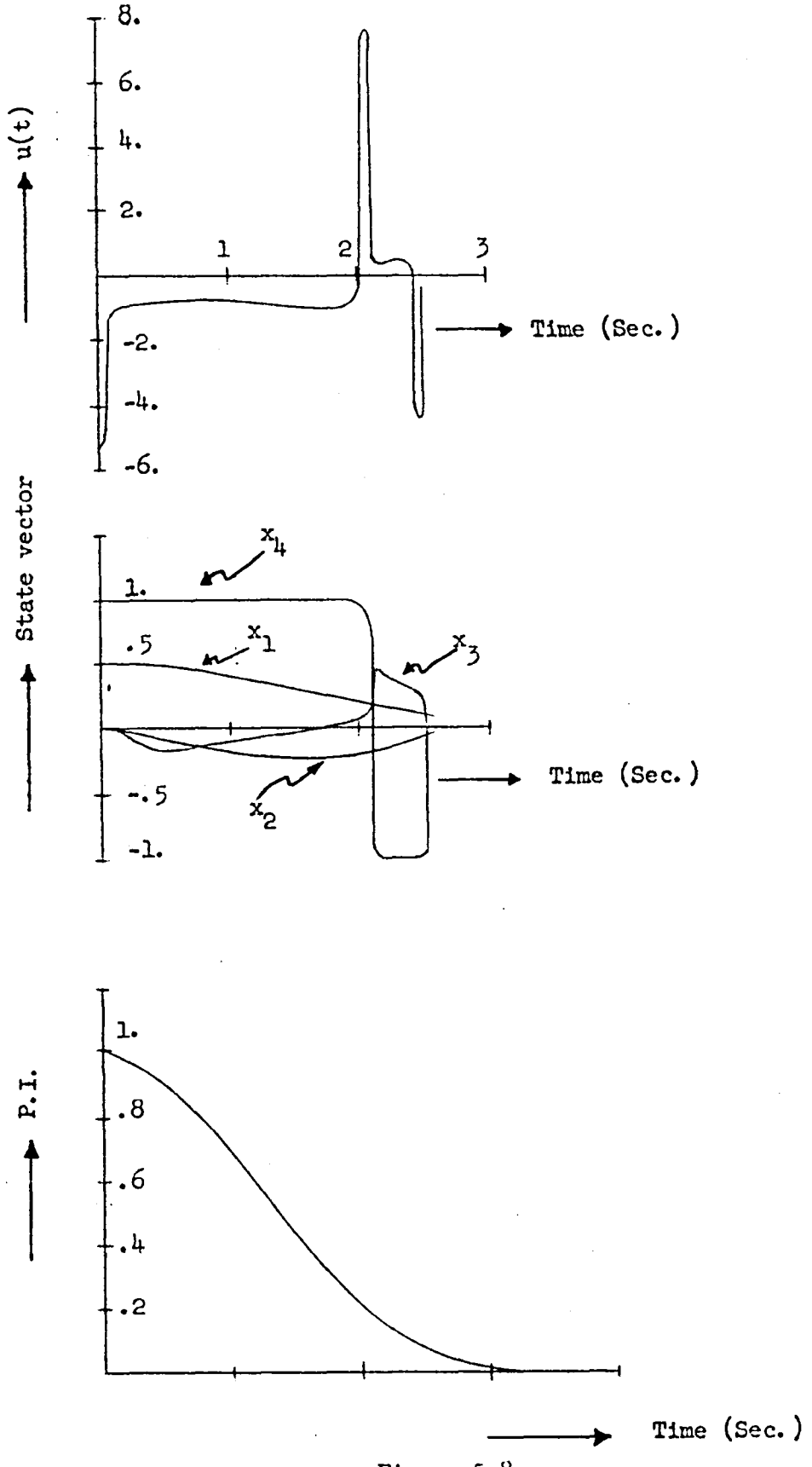


Figure 5.8

Ho and Brentani Solution to Example 5.2 with the Constraint $|x_4| \leq 1$.

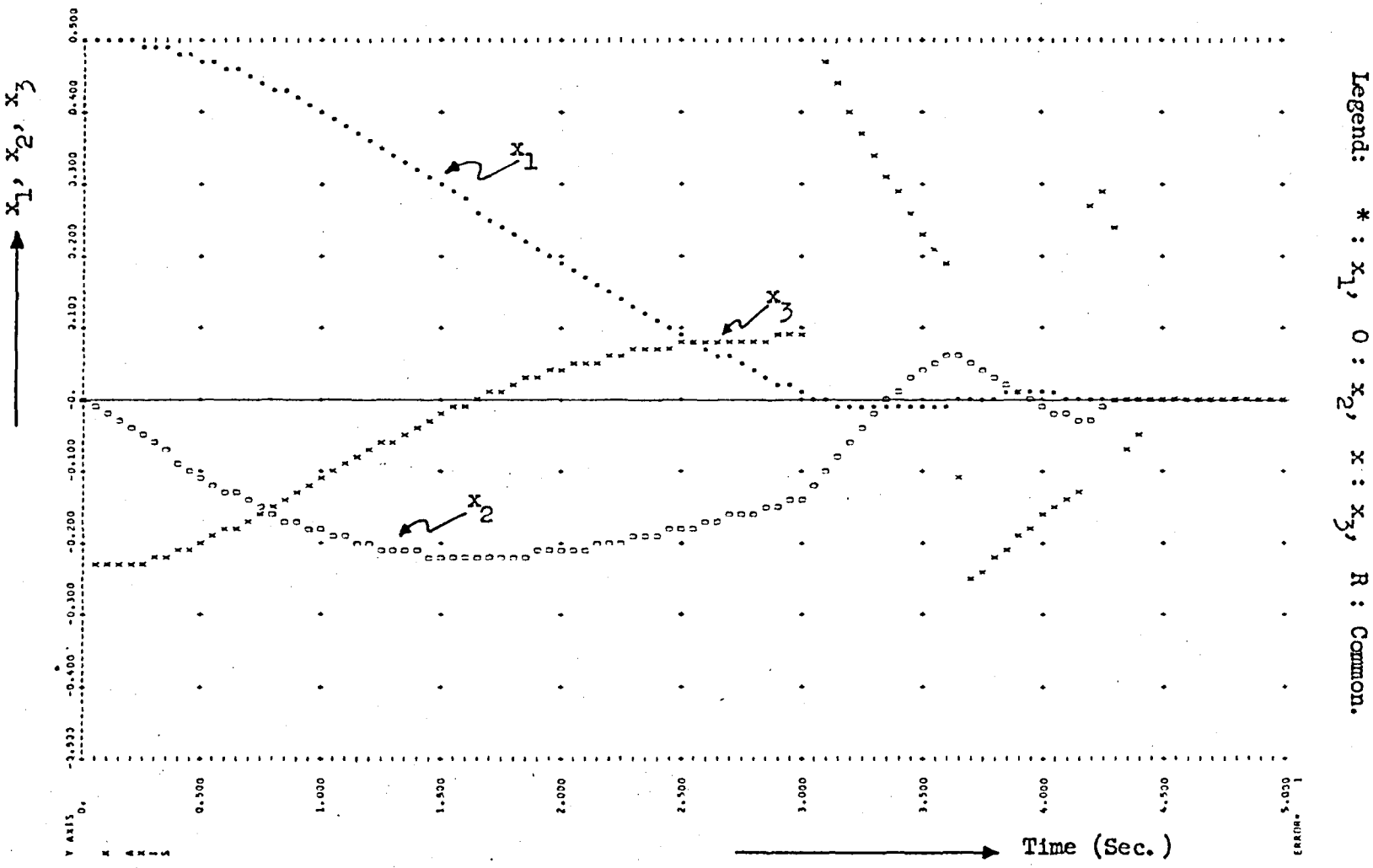


Figure 5.9 Solution to Example 5.2 under MOCP with the Constraint $|x_4| \leq 1$.

(a) x_1 vs. time, x_2 vs. time, x_3 vs. time.

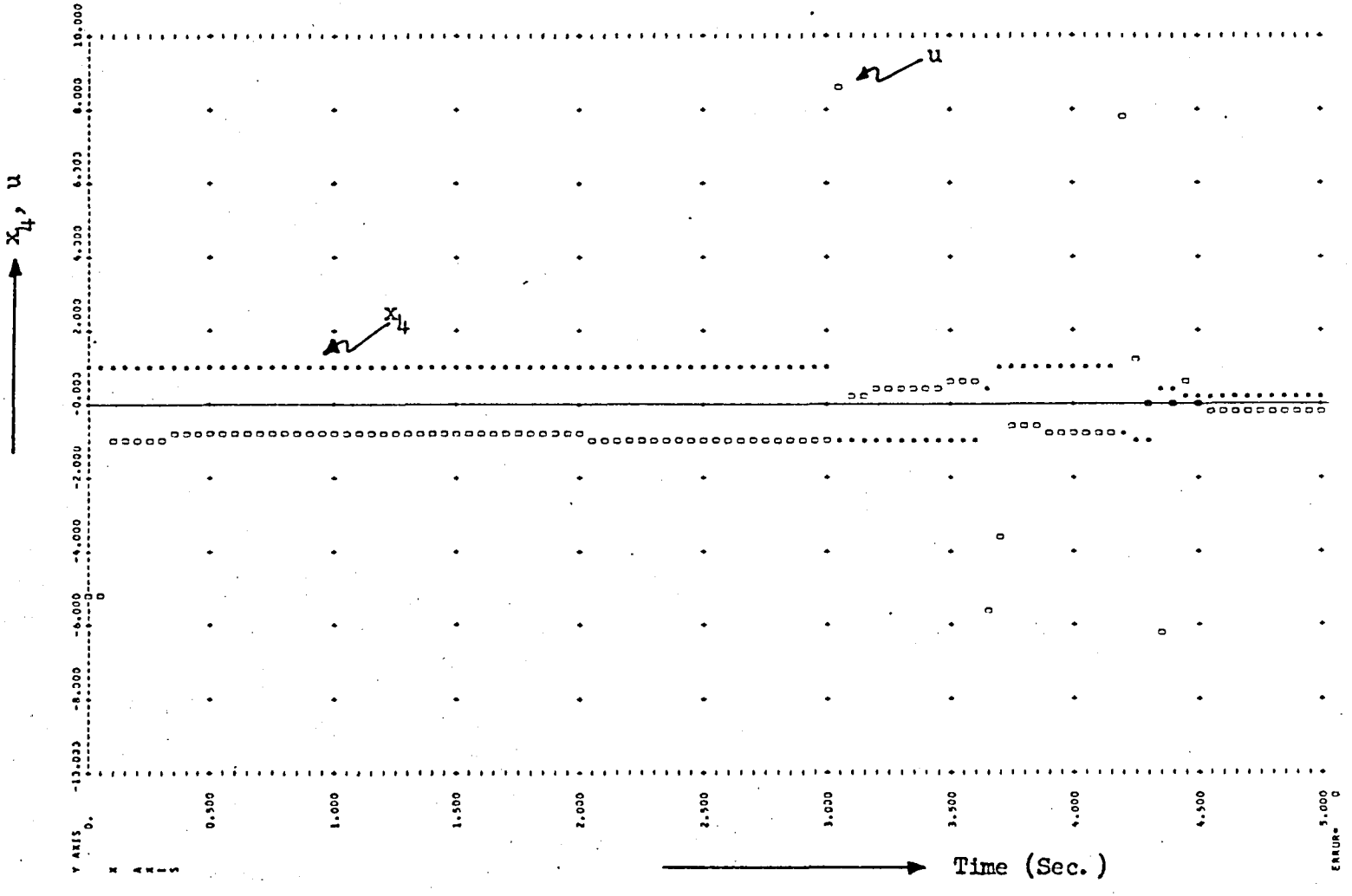


Figure 5.9 Solution to Example 5.2 under MOCP with the Constraint $|x_4| \leq 1$.

(b) x_4 vs. time, u vs. time.

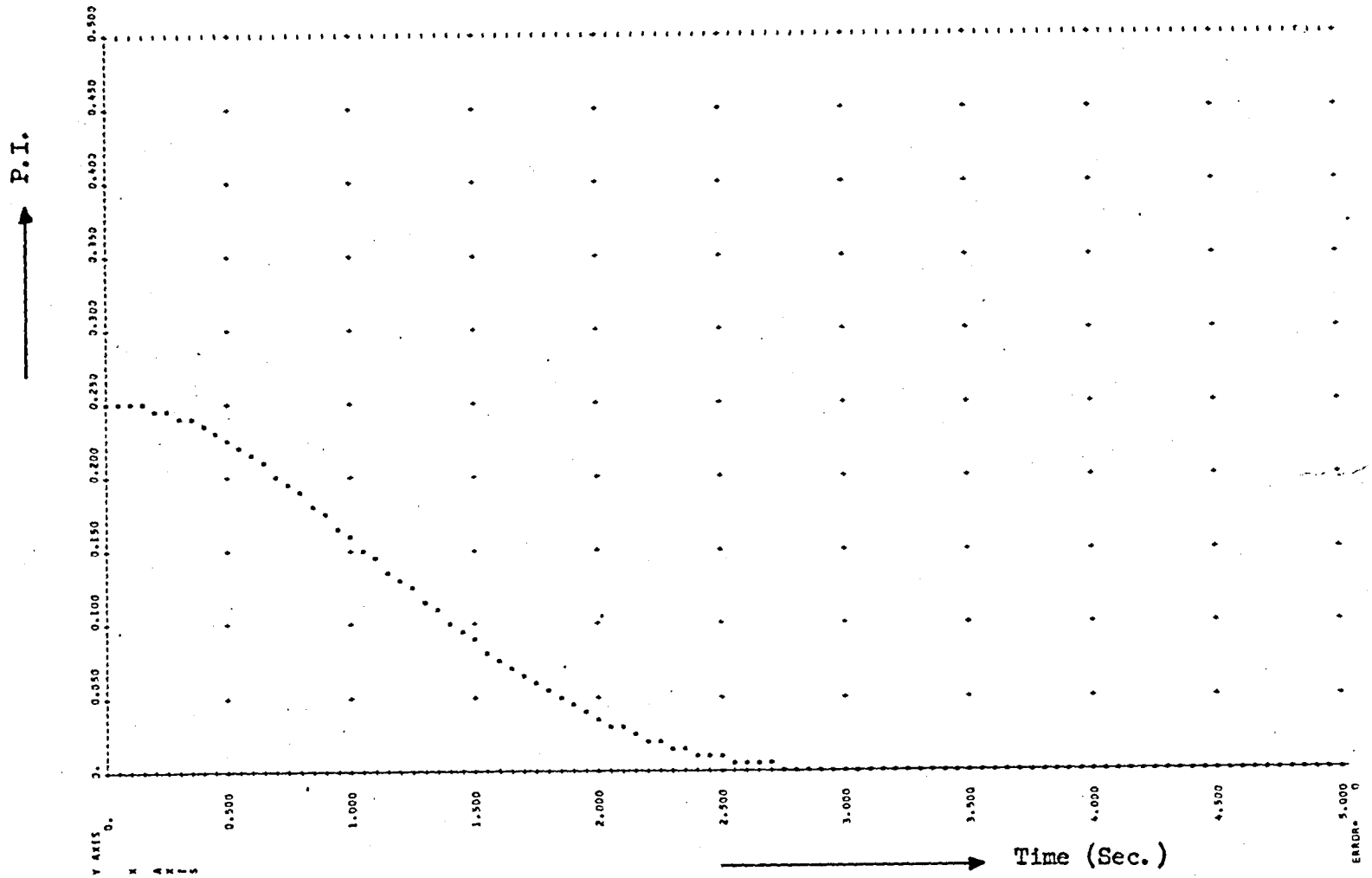


Figure 5.9 Solution to Example 5.2 under MOCP with the Constraint $|x_4| \leq 1$.

(c) P.I. vs. time.

and

$$\alpha = x_1 - c_7 x_3 + v \tag{5.63}$$

For an elapsed flight time of 80 sec., when the aerodynamic pressure is maximum, the values of the coefficients above are

$$\begin{aligned} c_1 &= -0.0791 & c_4 &= 10.93 \\ c_2 &= 0.448 & c_5 &= 21. \\ c_3 &= 5.59 & c_7 &= 0.00192 \end{aligned} \tag{5.64}$$

and the rigid body equations reduce to

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ .0791 & 0 & -.00153 \\ 26.59 & 0 & -.0108 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ -.448 \\ 10.93 \end{bmatrix} u + \begin{bmatrix} 0 \\ .0791 \\ 5.59 \end{bmatrix} v \tag{5.65}$$

$$\alpha = x_1 - 0.00192\, x_3 + v \tag{5.66}$$

The bending moment at the point where the first and the second booster stages meet, i.e., at 90 meters from bottom end, is given by

$$B.M. = 1.845x_1 - 0.00335x_3 + 12.38u + 1.845v \tag{5.67}$$

It should be emphasized here that the above equations are based on the assumption that the various angles involved are small which makes it possible to neglect the nonlinear terms.

In all the examples below, the initial conditions are assumed to be zero.

Example 5.3: Given the plant described by equations (5.65), (5.66) and (5.67), it is desired to minimize

$$P.I. = 5[\,|x_1(K+1)| + |x_1(K+2)|\,] + 0.02\,[\,|x_3(K+1)| + |x_3(K+2)|\,] \tag{5.68}$$

subject to the constraints

$$|u| \leq 0.0872 \text{ rad} = 5^{\circ}$$
$$|x_1| \leq 10^{\circ}$$
$$|\alpha| \leq 15^{\circ}$$
$$|\text{B.M.}| \leq 2.24 \times 10^{6} \text{ Kg.m.} \qquad (5.69)$$

x_1 and α are constrained, because if they get too large, the nonlinear terms neglected in the plant equations become significant and the mathematical model does not remain valid.

For a disturbance v of 10°, the solution for a period of 10 sec. is shown in Figure 5.10. The attitude angle remains negligibly small over the entire period. The drift rate is 12.3 meters/sec. at the end of 10 secs. and the total drift is 64.4 meters.

Decrease in the weight of the drift rate in the performance index of the above example does not have any noticeable effect on the system performance. However, when the coefficient of the drift rate in equation (5.68) is increased the solution becomes infeasible, because the constraint on α is violated. No longer does the attitude angle remain zero. In order to find how the system behaves over a period of 10 sec. when the weight of the drift rate is increased to 0.1, example 3 is solved with the new performance index (5.71), and without any state variable constraint, thus avoiding the infeasibility of the solution from the computational point of view.

Example 5.4: Given the plant of example 5.3 and the constraint

$$|u| \leq 0.0872 = 5^{\circ} \qquad (5.70)$$

minimize

$$\text{P.I.} = 5[\,|x_1(K+1)| + |x_2(K+2)|\,] + 0.1\,[\,|x_3(K+1)| + |x_3(K+2)|\,] \qquad (5.71)$$

The solution is given in Figure 5.11. Increase in the coefficient of x_3(i.e. $\dot{Z}$) does reduce the drift rate at first, however, the attitude

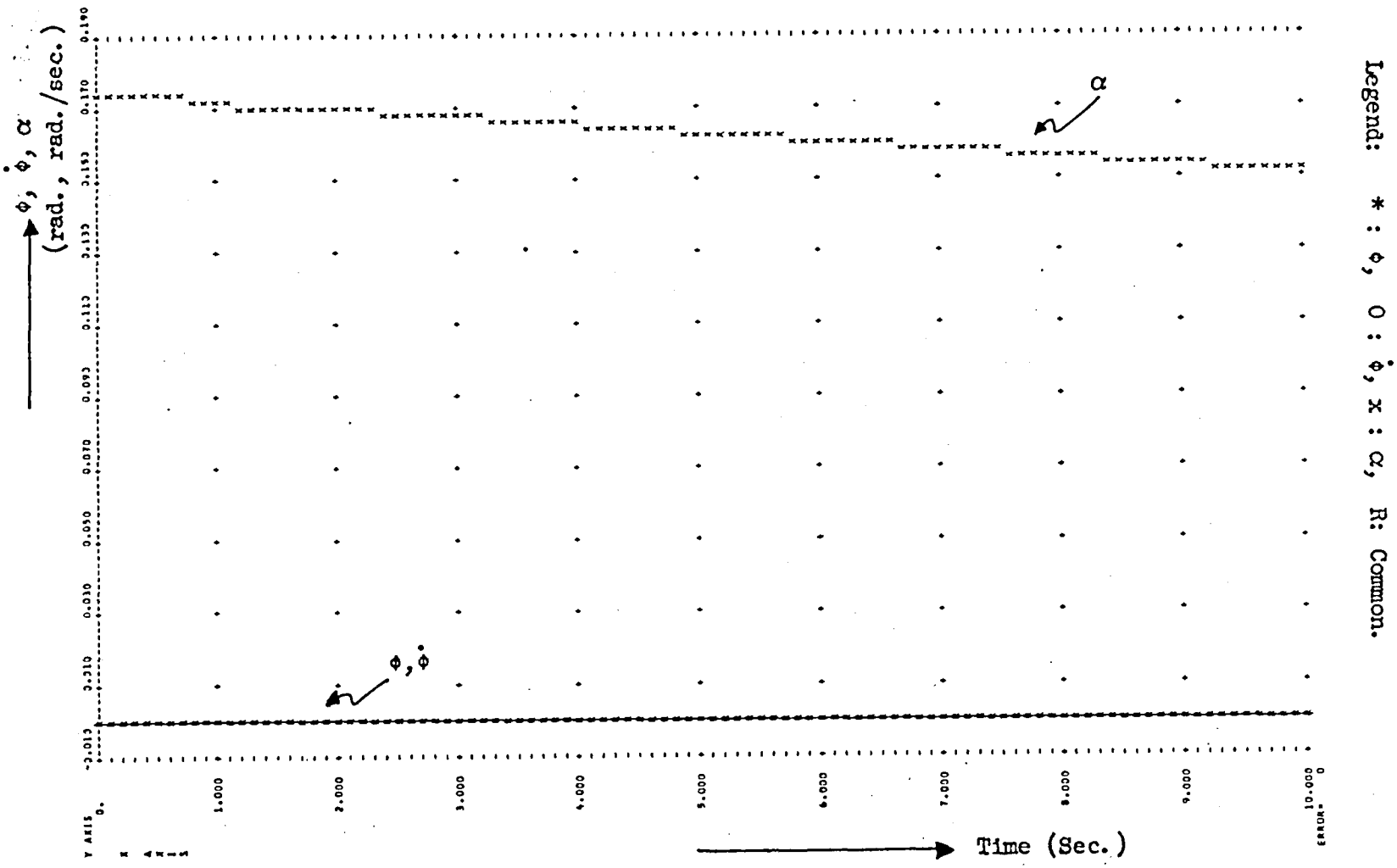


Figure 5.10 Response of MV2 (rigid, time-invariant) under MOCP with P.I. (5.68) and State and Control Variable Constraints (5.69).

(a) ϕ vs. time, $\dot{\phi}$ vs. time, α vs. time.

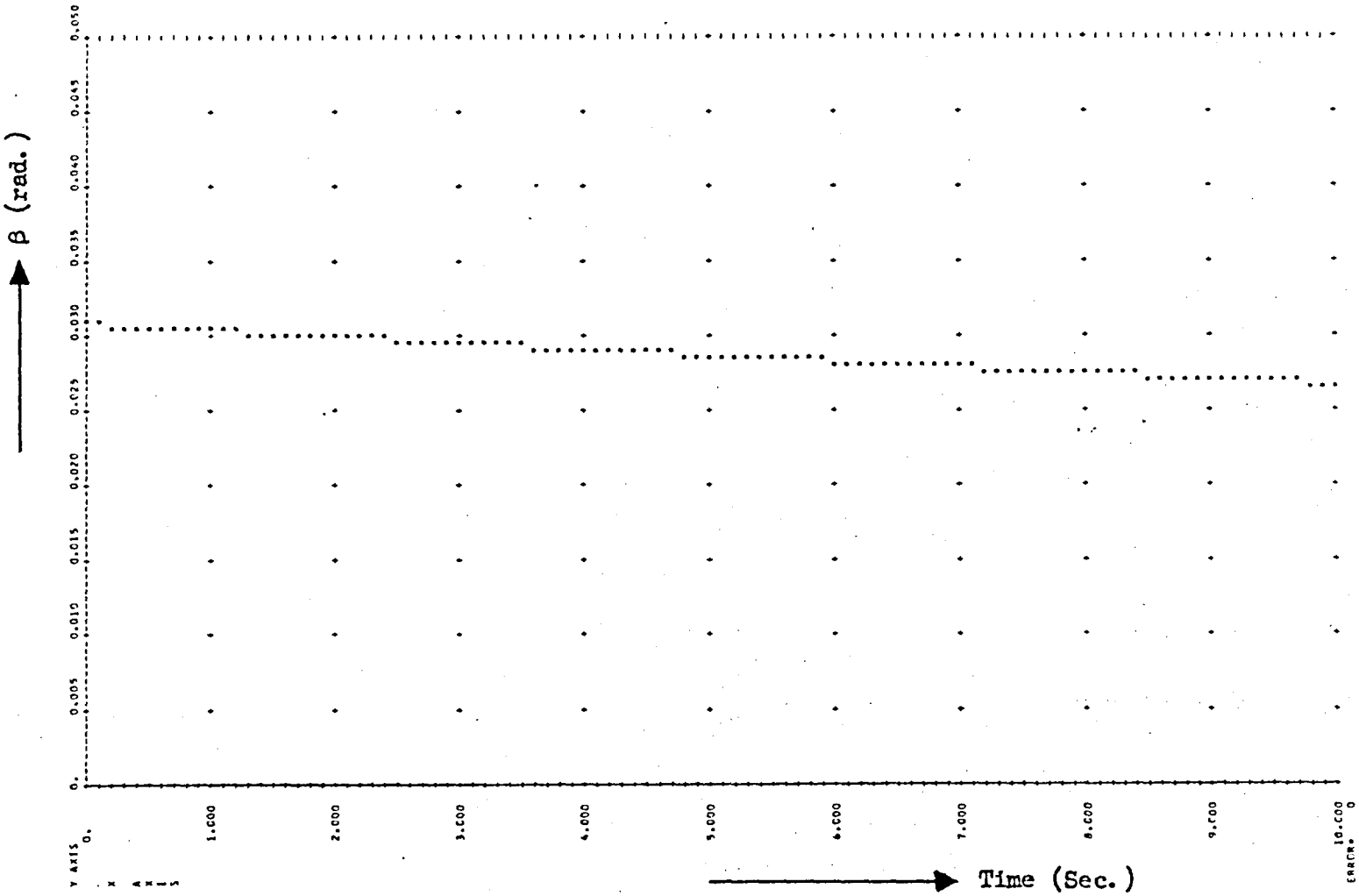


Figure 5.10 Response of MV2 (rigid, time-invariant) under MOCP with P.I. (5.68) and State and Control Variable Constraints (5.69).

(b) β vs. time.

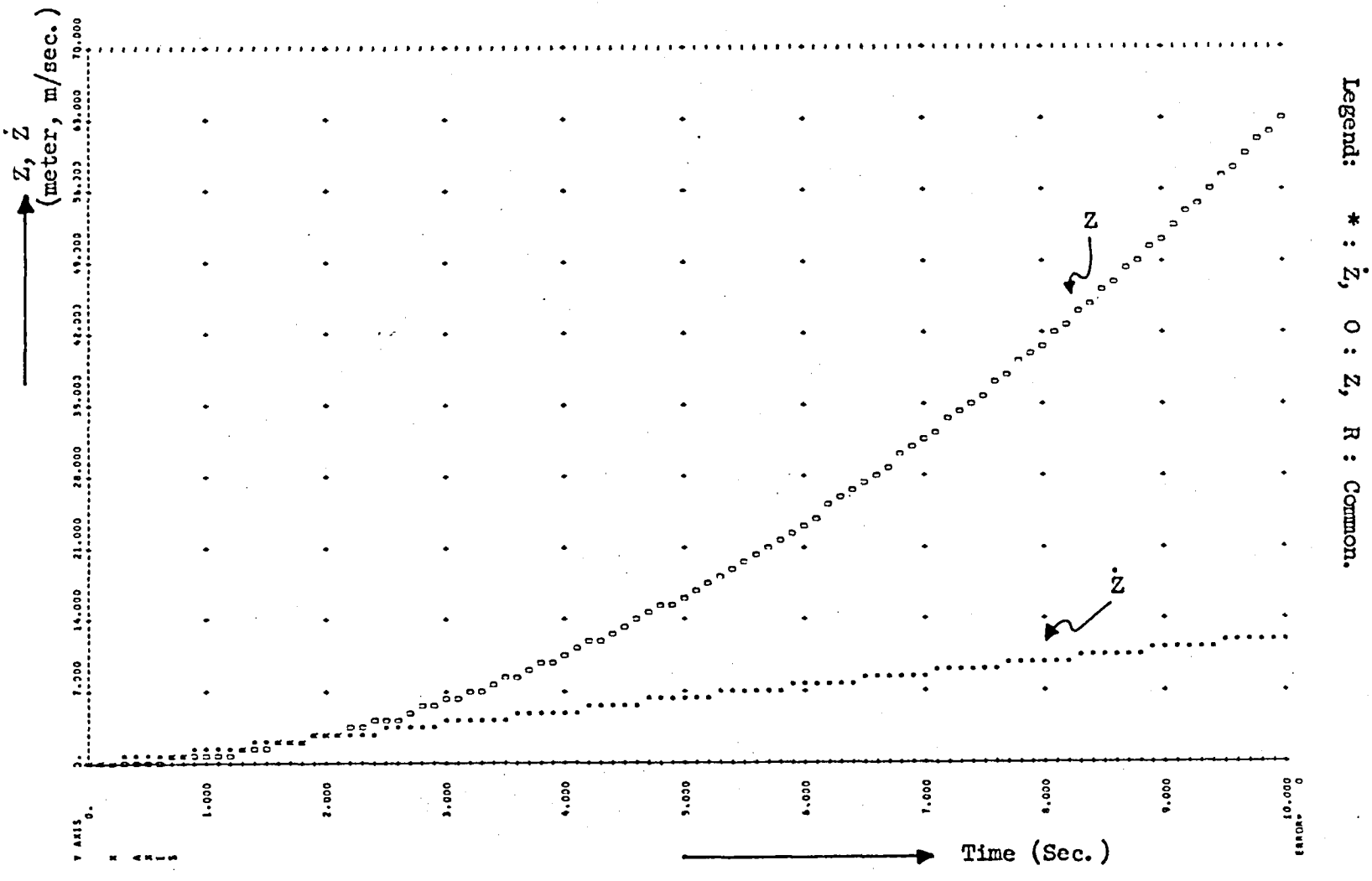


Figure 5.10 Response of MV2 (rigid, time-invariant) under MOCP with P.I. (5.68) and State and Control Variable Constraints (5.69).

(c) $\dot{Z}$ vs. time, Z vs. time.

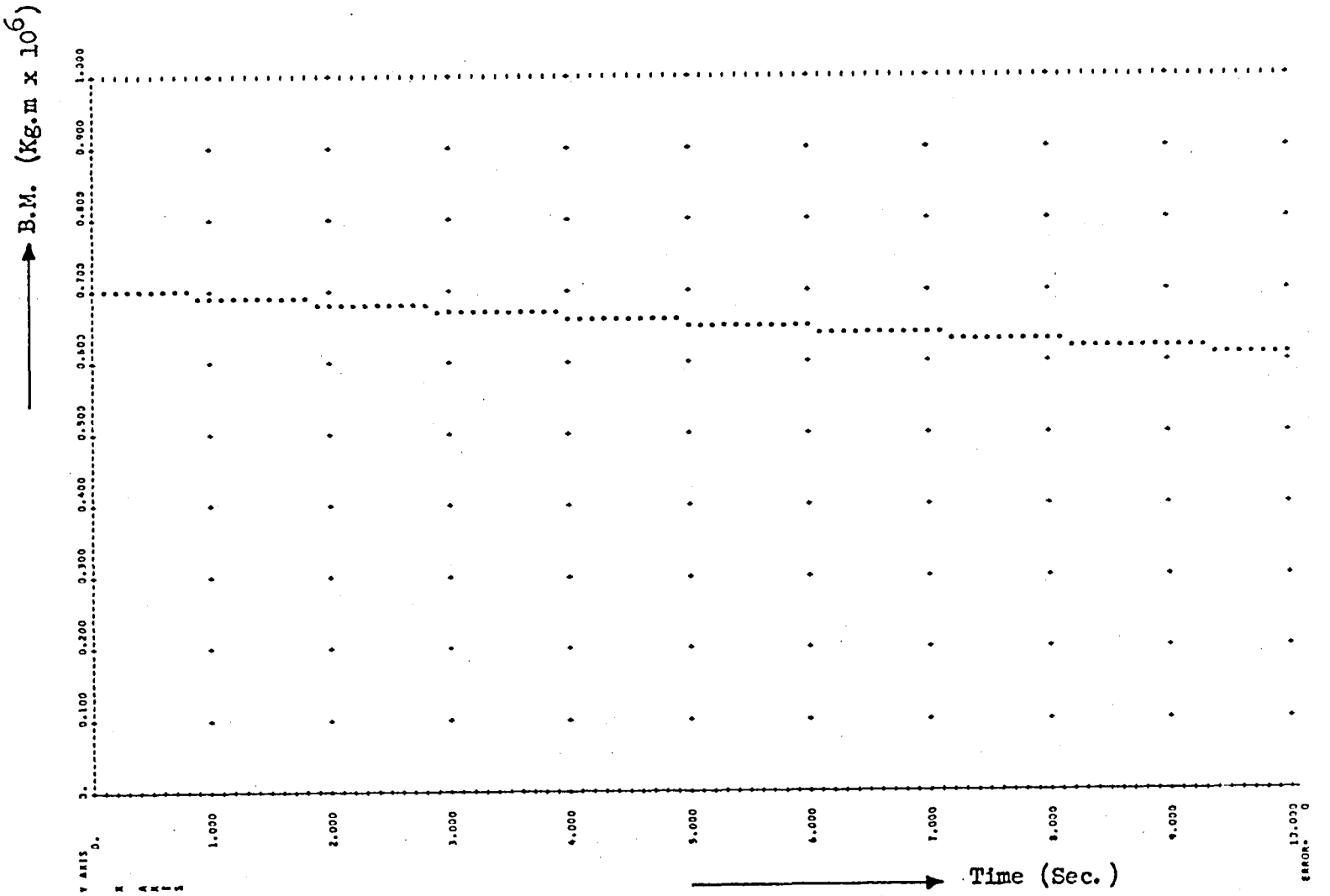


Figure 5.10 Response of MV2 (rigid, time-invariant) under MOCP with P.I. (5.68) and State and Control Variable Constraints (5.69).

(d) B.M. vs. time.

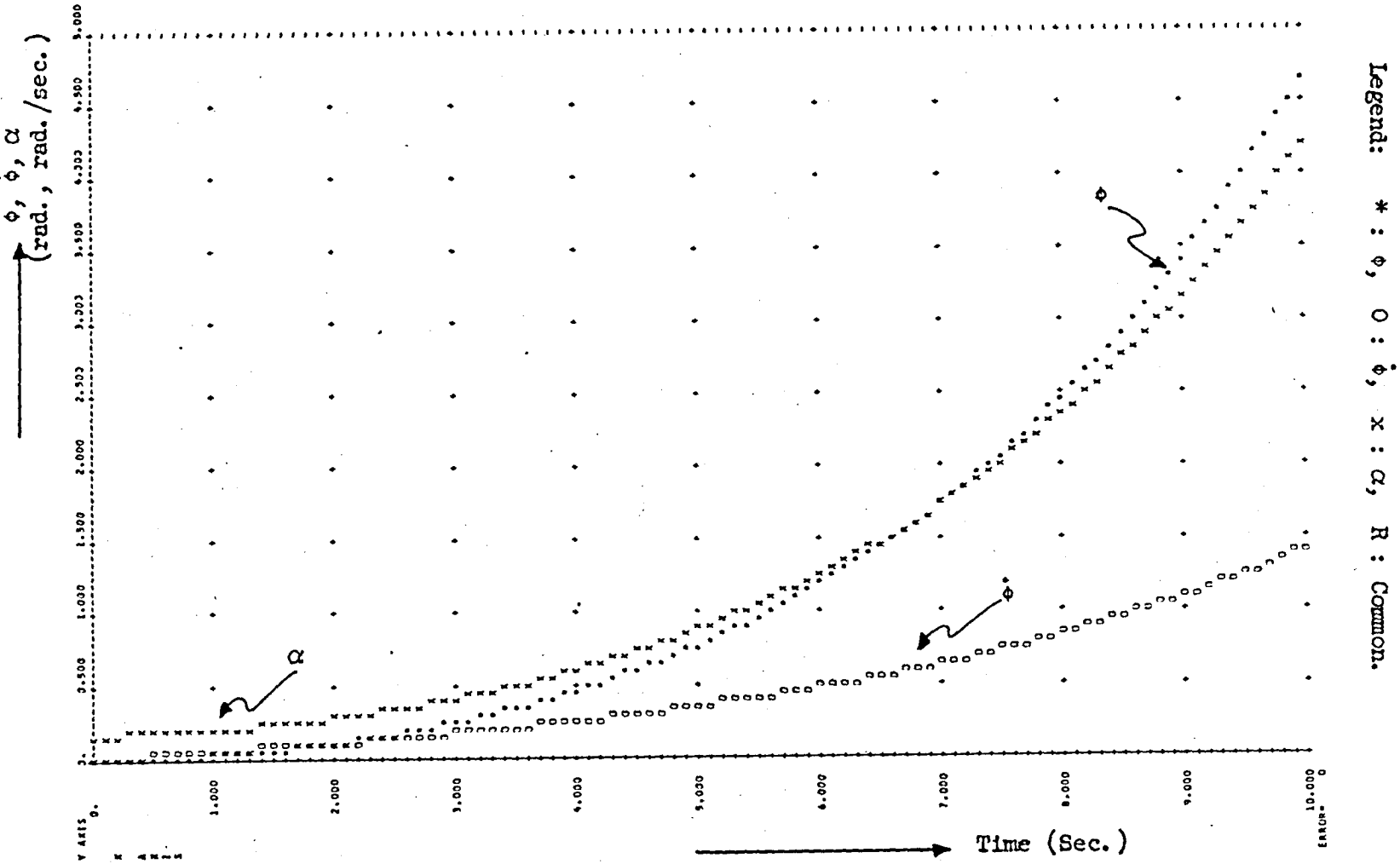


Figure 5.11 Response of MV2 (rigid, time-invariant) under MOCP with Performance Index (5.71) and Constraint (5.70).

(a) ϕ vs. time, $\dot{\phi}$ vs. time, α vs. time.

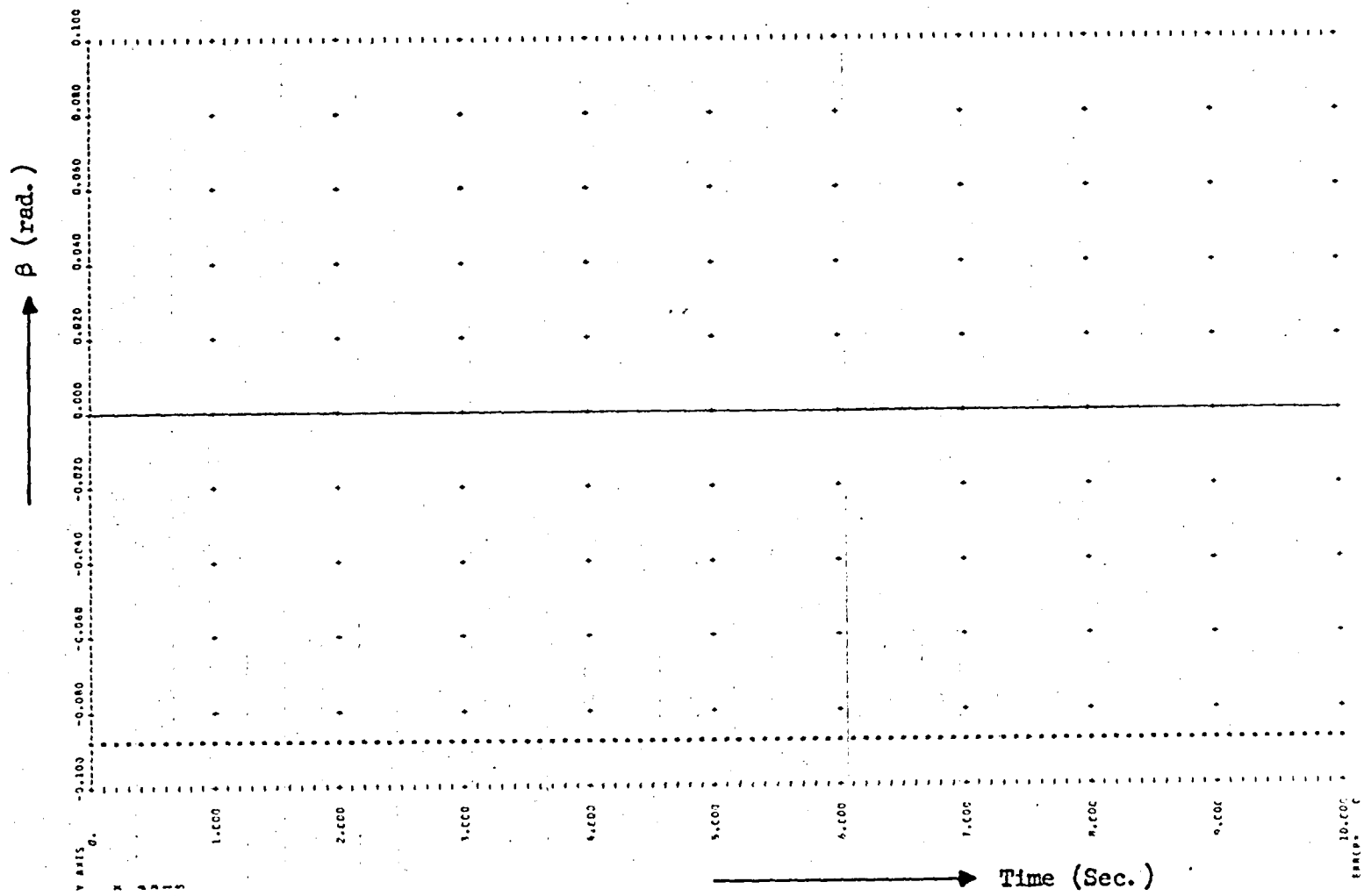


Figure 5.11 Response of MV2 (rigid, time-invariant) under MOCP with Performance Index (5.71) and Constraint (5.70).

(b) β vs. time.

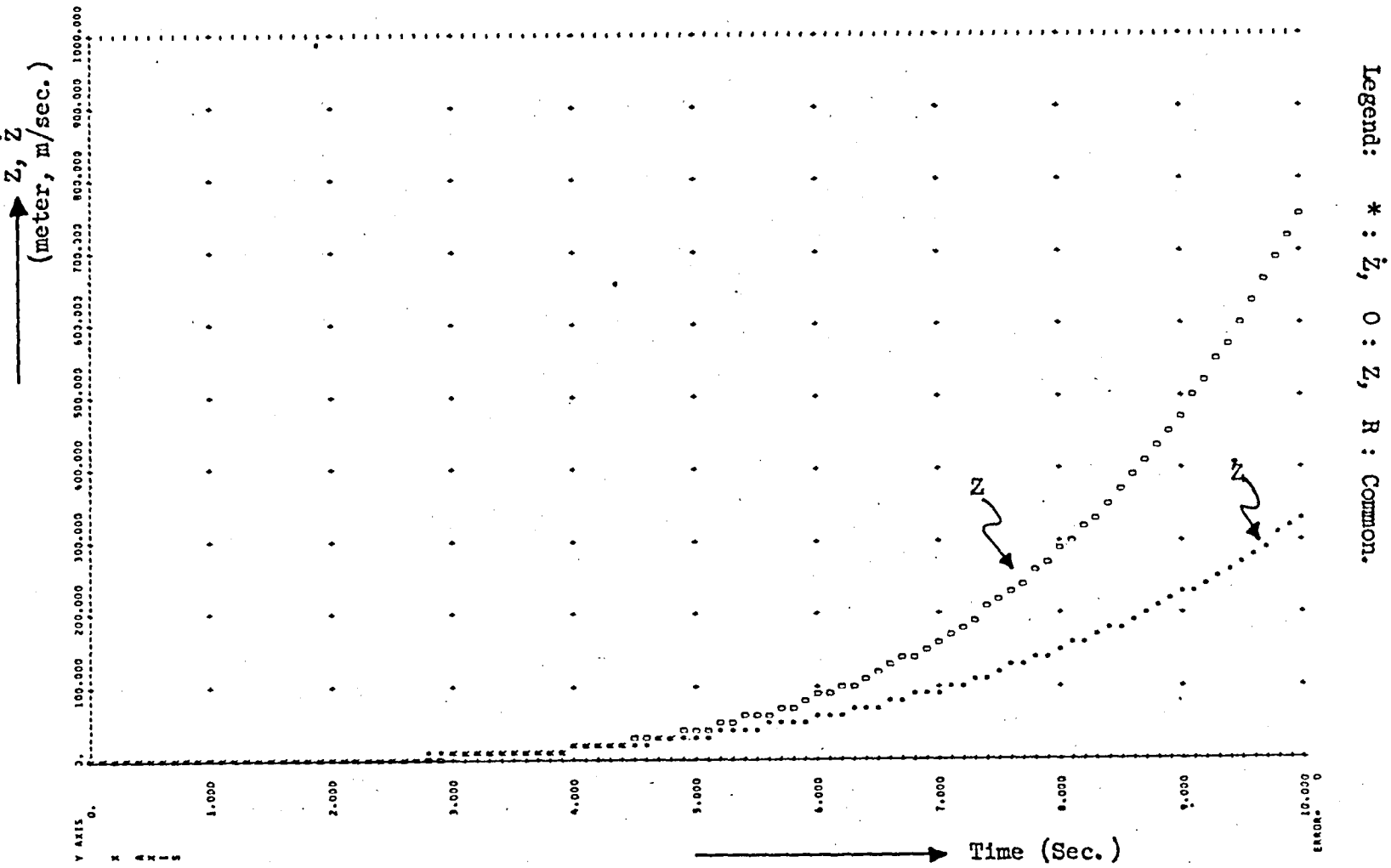


Figure 5.11 Response of MV2 (rigid, time-invariant) under MOCP with Performance Index (5.71) and Constraint (5.70).

(c) $\dot{Z}$ vs. time, Z vs. time.

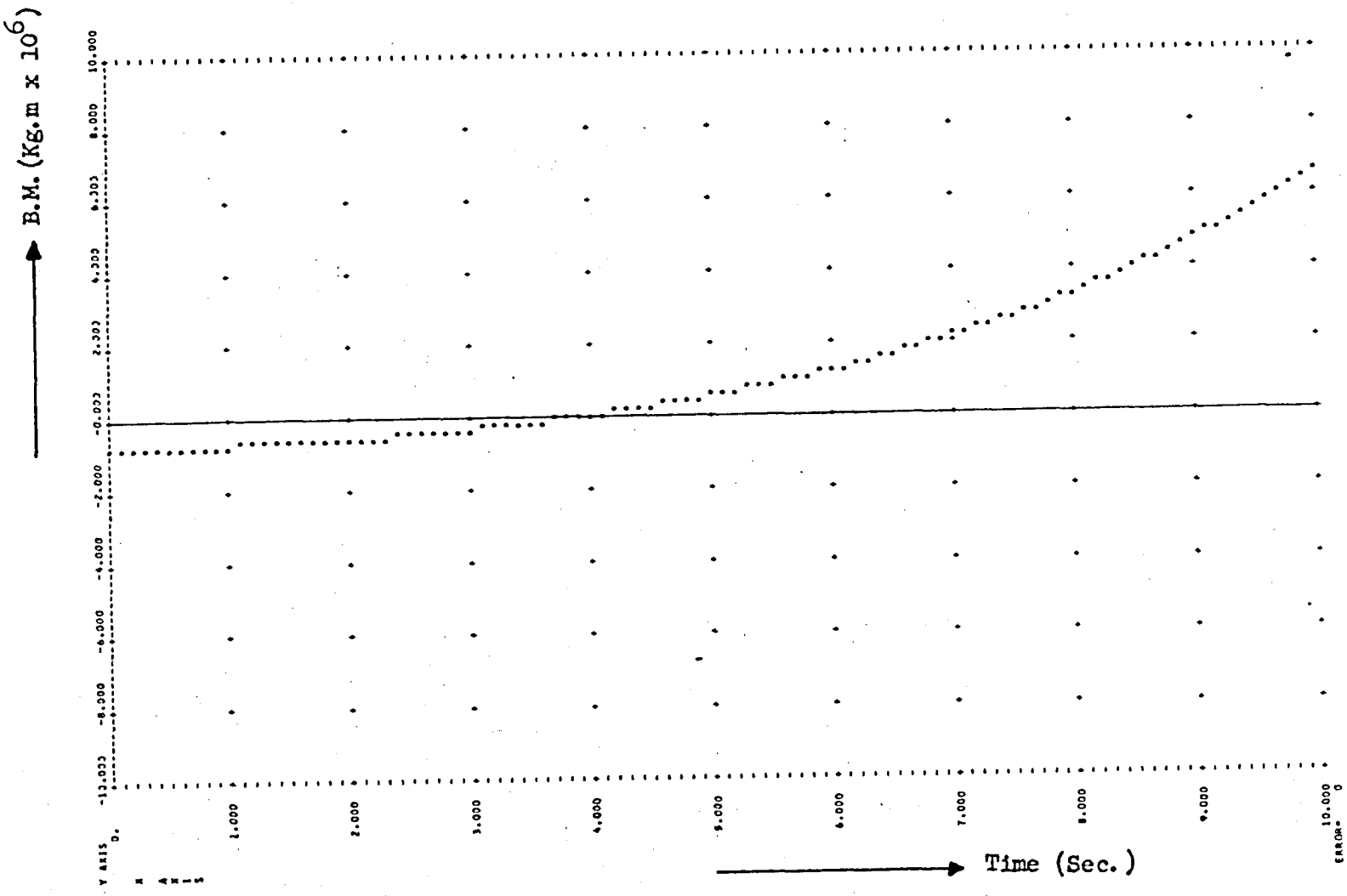


Figure 5.11 Response of MV2 (rigid, time-invariant) under MOCP with Performance Index (5.71) and Constraint (5.70).

(d) B.M. vs. time.

angle does not remain zero. As time goes on, increase in the attitude angle contributes to $\ddot{Z}$ as can be seen from equation (5.59); thus drift rate starts building up. At the end of 10 seconds, the drift becomes as large as 787 meters and the attitude angle and the angle of attack well over four radians. It is concluded that this performance index under the present control policy is meaningless; no further investigations are made with this performance index.

In both examples 5.3 and 5.4 the bending moment is well below the design limits. However, in order to study the system behaviour when the bending moment is minimized, example 5.5, where the performance index is a function of the bending moment, is solved.

Since the control signal u(K) is piecewise constant, the bending moment can take four different values over two intervals, namely

$$BM_1(t_K^+) = 1.845x_1(K) - .00355x_3(K) + 12.38u(K) + 1.845v(K) \qquad (5.72)$$

$$BM_2(t_{K+1}^-) = 1.845x_1(K+1) - .00355x_3(K+1) + 12.38u(K) + 1.845v(K+1) \qquad (5.73)$$

$$BM_3(t_{K+1}^+) = 1.845x_1(K+1) - .00355x_3(K+1) + 12.38u(K+1) + 1.845v(K+1) \qquad (5.74)$$

$$BM_4(t_{K+2}^-) = 1.845x_1(K+2) - .00355x_3(K+1) + 12.38u(K+1) + 1.845v(K+2) \qquad (5.75)$$

The sum of the absolute values of the above bending moment is minimized in example 5.5.

Example 5.5: Given the plant of example 5.3 and the constraint (5.70), minimize

$$P.I. = |B.M._1| + |B.M._2| + |B.M._3| + |B.M._4| \qquad (5.76)$$

As was observed with the drift rate in example 5.4, here (Figure 5.12) the bending moment is, for the first 5 secs., much smaller than in example 5.3. However, the attitude angle builds up consistently and the drift rate is large as compared to that in example 5.3. After about 6 secs. the term containing the attitude angle in the bending moment expression becomes dominant over the term containing the drift rate. As a result bending

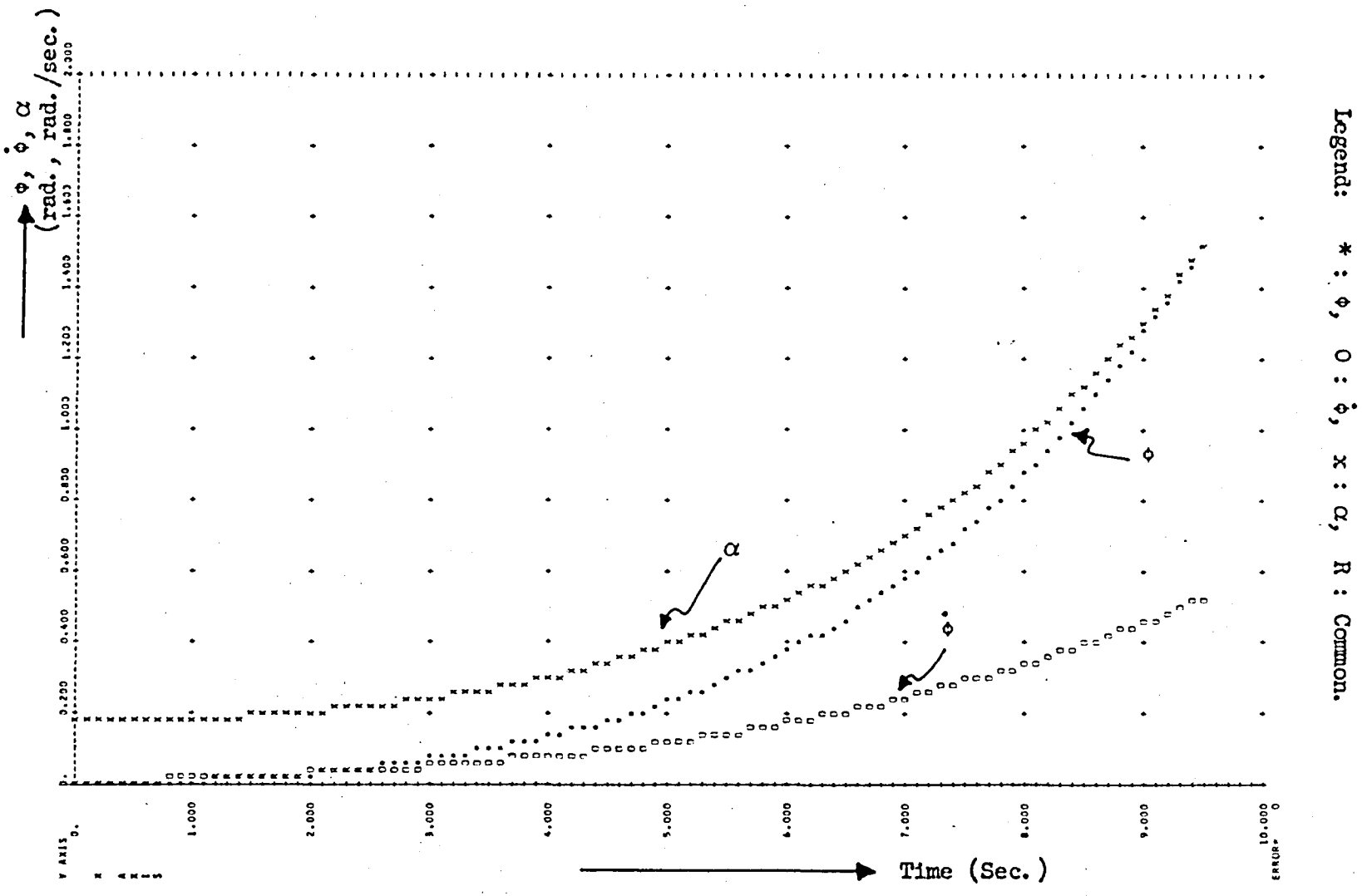


Figure 5.12 Response of MV2 (rigid, time-invariant) under MOCP with P.I. (5.76) and Constraint (5.70).

(a) ϕ vs. time, $\dot{\phi}$ vs. time, α vs. time.

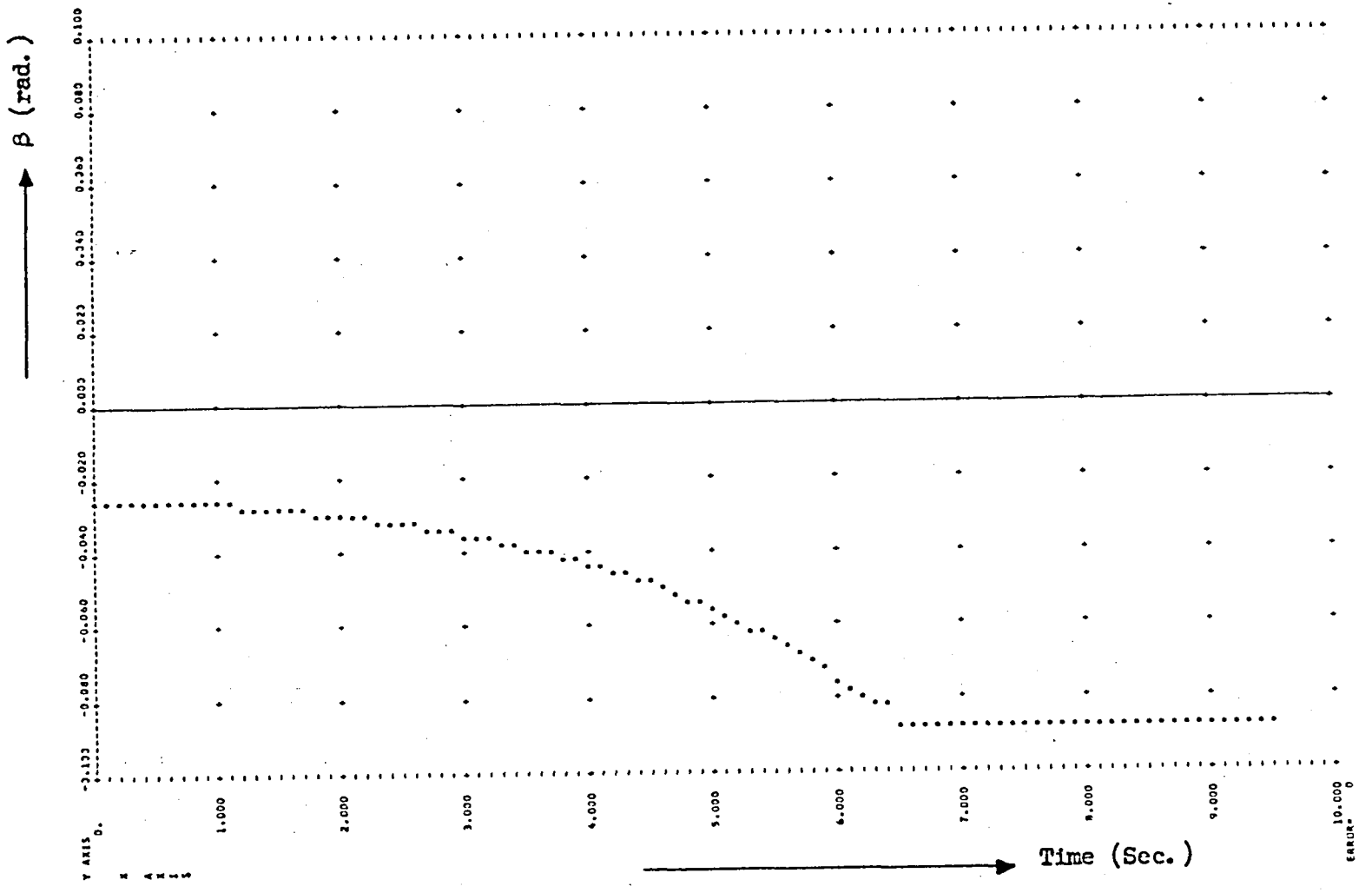


Figure 5.12 Response of MV2 (rigid, time-invariant) under MOCP with P.I. (5.76) and Constraint (5.70).

(b) β vs. time.

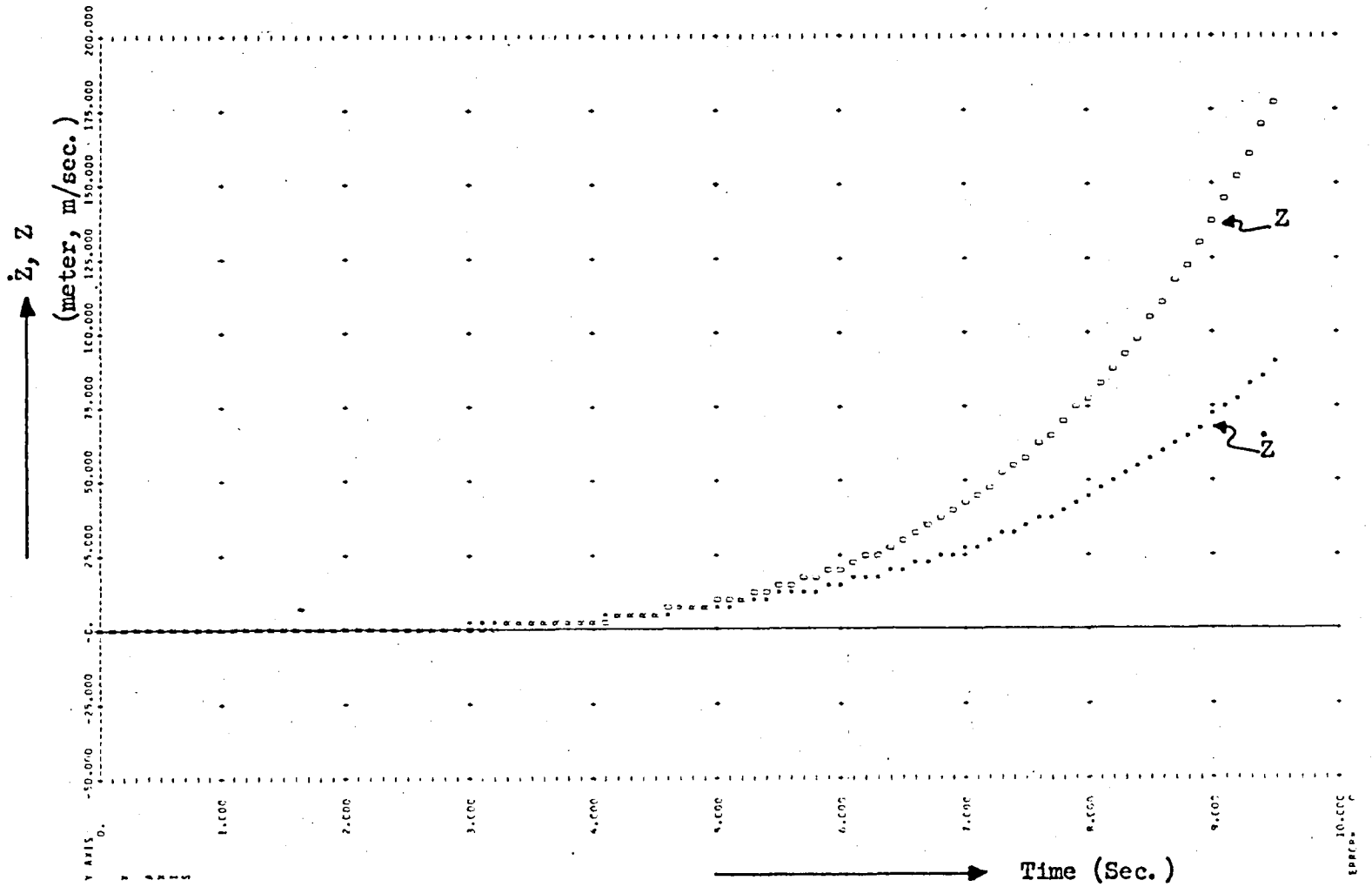


Figure 5.12 Response of MV2 (rigid, time-invariant) under MOCP with P.I. (5.76) and Constraint (5.70).

(c) $\dot{Z}$ vs. time, Z vs. time.

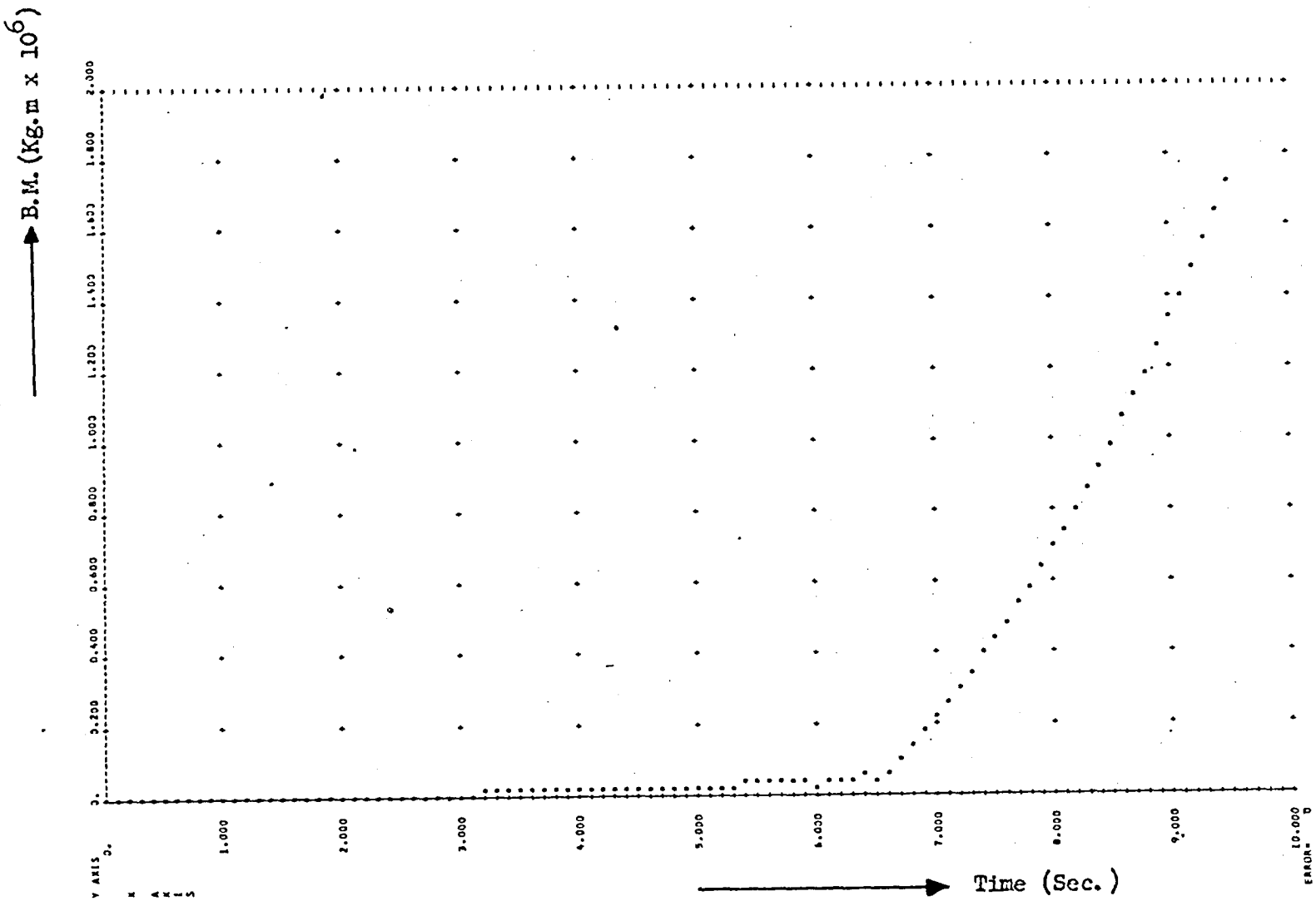


Figure 5.12 Response of MV2 (rigid, time-invariant) under MOCP with P.I. (5.76) and Constraint (5.70).

(d) B.M. vs. time.

moment increases rapidly. At the end of 10 secs. the bending moment becomes about 150 % of the design limit.

From the point of view of staying close to the reference trajectory and well within the design specifications, the performance index in equation (5.68) is found to yield the best results. Using this performance index a solution for the time varying system, over the trajectory time of 66.5 to 89.5 secs. with the disturbance v(i.e.,α_w) provided by M.S.F.C., is also found. This is given in the following example.

Example 5.6: For the time-varying plant described by equations (5.62) and (5.63) and the disturbance v(i.e.,α_w) in Figure 5.13, minimize the right hand side of equation (5.68) subject to equation (5.69).

The solution is given in Figure 5.14. From the computational point of view, this example is not different from example 5.3. The time-varying plant can be handled as easily as the time-invariant plants. Here as in example 5.3, the attitude angle is zero over the entire trajectory and the drift rate at the end of 23.5 seconds period is 15.3 meters/sec. The bending moment is well below the design limit.

5.8 Conclusions and Recommendations

The optimal digital control problem related to linear time invariant or time-varying systems with absolute value type performance index, and state and control variables constraints, has been reduced to the standard form of linear programming. In order to avoid prohibitively large memory capacity requirements for solution of such problems, a new concept of modified optimal control policy is introduced. Necessary manipulations, based on series approximation required for reduction of a continuous system to a discrete system are given. The type of problems considered for linear programming formulation include one-interval control and two-interval control problems with ordinary and modified optimal control policy. Also,

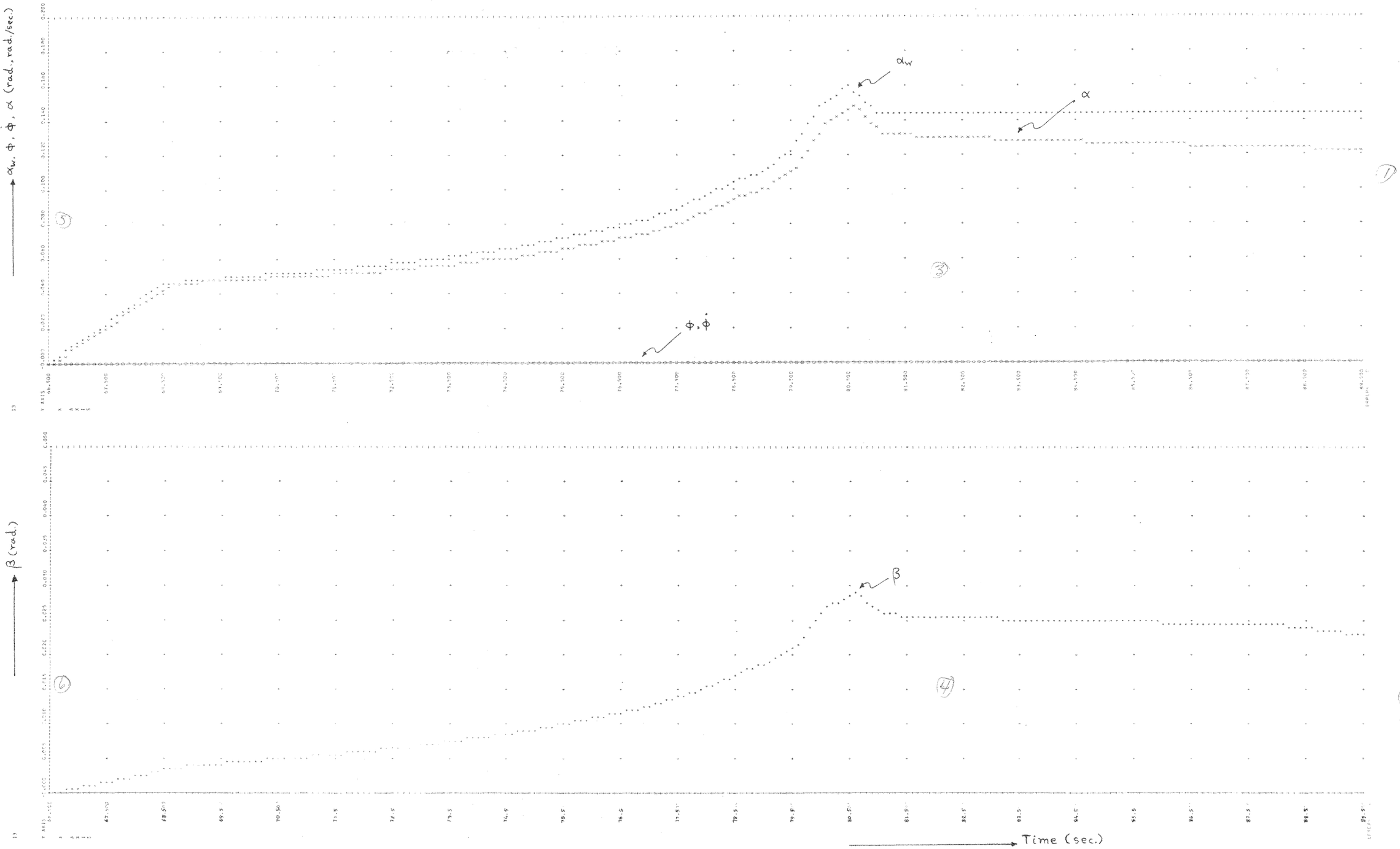


Figure 5.13 Response of MV 2 (rigid, time-varying) under MOCP with performance index (5.68) and constraints (5.69).
(a) α_w vs Time, ϕ vs Time, $\dot{\phi}$ vs Time, α vs Time, β vs Time.

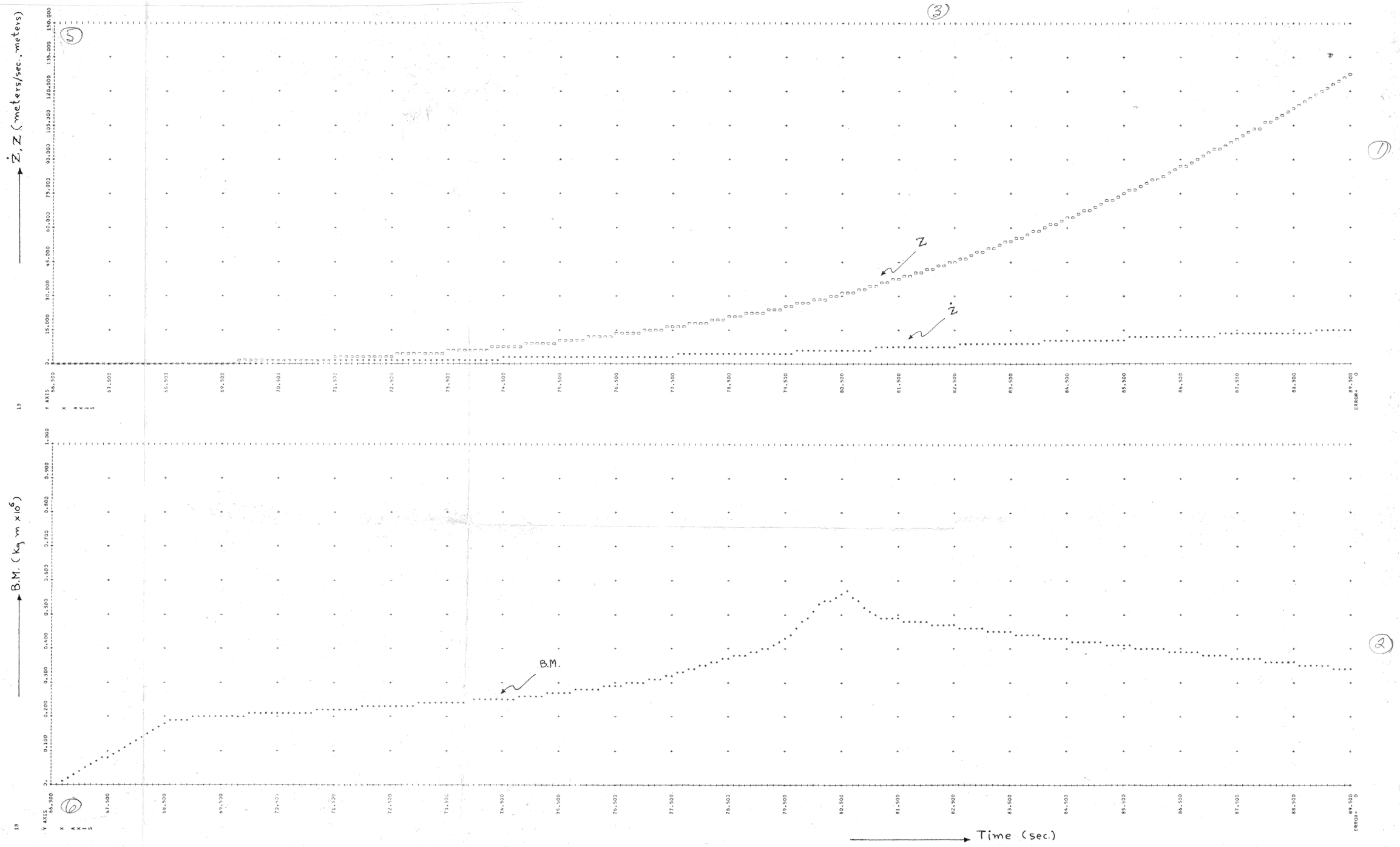


Figure 5.13 Response of MV 2 (rigid, time-varying) under MOCP with performance index (5.68) and constraints (5.69).
(b) ż vs Time, z vs Time, BM vs Time.

N-interval control problem and the problem of minimizing the maximum deviation is considered. Through examples, it is demonstrated that optimization under the modified optimal control policy, when computational feasibility is also a point of consideration, yields better results. Solutions to both time-invariant and time-varying rigid body approximations to Model Vehicle No. 2 are obtained.

The concept of the modified optimal control policy is the major contribution of this chapter. The potential of this concept is not fully explored here; possible avenues for further research under this policy are given below.

The application of the modified optimal control policy is not limited to the linear programming technique of optimization alone. In the optimization problems considered here, only the control function is determined; no effort is made to determine a control law. However, under this policy, using dynamic programming with a quadratic performance index, an optimal control law can be obtained.

Optimization via calculus of variations or the maximum principle often leads to two-point-boundary-value-problems. It is felt that the solution of this two-point boundary value problem is easier under modified optimal control policy than otherwise. This needs to be confirmed.

In case of state and control variable constraints, it is not easy to use dynamic programming, the maximum principle or gradient techniques for optimization. Also with these schemes, except in a few trivial cases, it is impossible to obtain a closed loop solution. However, linear programming can be used to find computer solutions when the performance index is of the type expressed in equation (5.41).

Using linear programming, for a given value of $\underline{x}(K)$, $u(K)$ is determined. So using an on-line computer, what is essentially a closed-loop solution is found. For a successful closed-loop operation, the following problems need to be investigated.

(a) Figure 5.14 shows an on-line computer-controlled closed-loop system. Assume that the quantization period is determined by the time required for measurement of the states and the computation of the control signal. Evidently this system is not realizable since the computer and the controller can not generate $u(K)$ given $x(K)$ instantaneously. A definite time is required to compute and adjust the controller. Now since the plant is assumed to be deterministic it is possible to compute $\underline{x}(K)$ knowing $u(K-1)$.

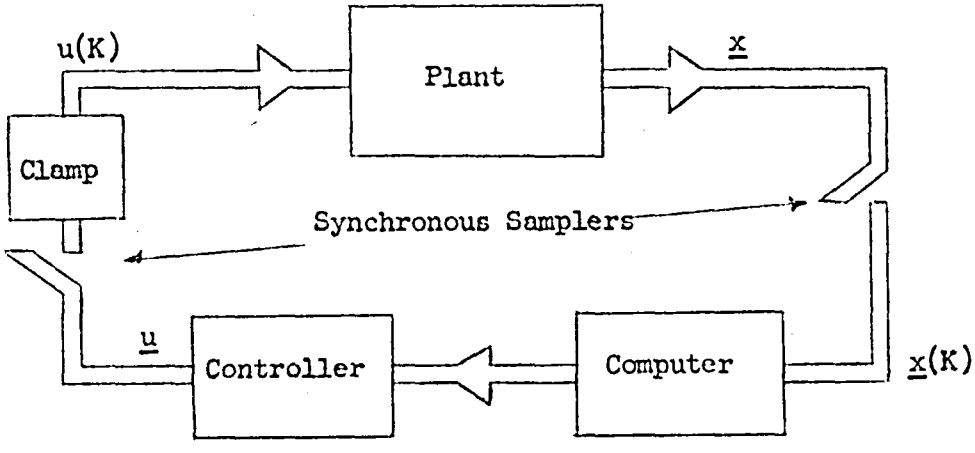


Figure 5.14

Computer-Controlled Closed-Loop System

This means we do not have to measure $\underline{x}(K)$ to calculate $u(K)$. This argument is all right only if there are no disturbances, otherwise the computed value of the state, $\underline{x}(K)$ will not be the same as the measured values of the states, $\hat{\underline{x}}(K)$. If $\underline{x}(K) \neq \hat{\underline{x}}(K)$ what scheme should be used for computer control? Is the use of equation (5.77) instead of equation (5.9) a practical solution to the problem?

$$\underline{x}(K+1) = A(t_{K+1} - t_K)\,\underline{x}(K) + h(K)\,u(K) + A(t_{K+1} - t_{K-1})[\hat{x}(K-1) - x(K-1)] \quad (5.77)$$

Or is it possible to estimate $\hat{x}(K)$ knowing $\underline{x}(K)$ and the stochastic properties of the disturbances?

If the time required to compute the optimal control signal for a discrete system happens to be smaller than the sampling period, yet not small enough to be negligible, a similar problem exists. Notice that the standard assumption made in sample data theory is that computational time is negligible.

(b) During on-line control if any of the constraints exceed the limits, the computer is no longer able to control the process. What precautions should be taken so that this does not happen? And if it does happen what alternative control policy should be used?

(c) As the order of the plant and the value of M(see equation(5.41)) increase, the number of variables and constraints using linear programming formulation becomes large. For a given n and M is it possible to cut down the number of variables and equations?

Following the work on linear systems, nonlinear systems should be studied under modified optimal control policy in conjunction with various optimization techniques, namely, non-linear programming, steepest descent methods, the maximum principle and linear programming together with quasilinearization.

Finally in the problems involving large amounts of computational work, there is often a possibility of truncation error. This error becomes significant if the total period of interest is large compared to the period of optimization under modified optimal control policy. This matter needs the attention of a numerical analyst. In appendix C, a suggestion is made to improve the accuracy of existing subroutines. This, however, requires some additional computation time.

REFERENCES

1. Zadeh, L. A., "On Optimal Control and Linear Programming," IRE Trans. on Automatic Control, Vol. AC-7, No. 4, July 1964, Correspondence.

2. Whalen, B. H., "On Optimal Control and Linear Programming," IRE Trans. on Automatic Control, Vol. AC-7, No. 4, July, 1964, Correspondence.

3. Whalen, B. H., "Linear Programming for Optimal Control, Ph.D. Dissertation, University of California, Berkeley, California, 1963.

4. Porcelli, G. and Fegley, K. A., "Linear Programming Design of Digitally Compensated Systems," Pre-prints of JACC Conference, 1964, Stanford University, Stanford, California.

5. "Optimization of Discrete Control Systems Through Linear Programming," Journal of the Franklin Institute, Vol. 278, Number 1, July 1964.

6. Dantzig, G. B., Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey, 1962.

7. Ho, Y. C., and Brentani, P. B., "On Computing Optimal Control with Inequality Constraints," Journal of the Society for Industrial and Applied Mathematics on Control, Series A, Vol. 1, Number 3, 1963.